



ניסוי מעבדה:

מקלט/משדר SDR, אלגוריתמים ובדיקת ביצועים

ניסוי בחירה מספר 071

במסגרת המקצוע "מעבדות 2-3 בחשמל"

כתב: אלכס הודיסן

הבהרה: חוברת זו מיועדת לביצוע ניסוי במעבדה לתקשורת בטכניון בלבד. אין להעתיק או להוציא או להפיץ או לפרסם חוברת זו בכל דרך או אמצעי מחוץ לטכניון



תוכן העניינים

3.....	תמצית הניסוי
5.....	פרק 1 – ציוד הניסוי
7.....	פרק 2 – רקע תאורטי
7.....	חלק א' – ערוץ השידור
8.....	חלק ב' – ערוץ הקליטה
10.....	חלק ג' – PLL דיגיטלי
12.....	חלק ד' – CFC
14.....	חלק ה' – FFC
17.....	חלק ו' – Timing Recovery
20.....	1. TED
21.....	a. גלאי שגיאת תזמון מסוג Maximum Likelihood Timing Error Detector (MLTED)
22.....	b. גלאי שגיאת תזמון מסוג Early-Late Timing Error Detector (ELTED)
22.....	c. גלאים נוספים
22.....	2. מסנן החוג $F(z)$
23.....	3. אינטרפולטור
23.....	a. אינטרפולציה פולינומיאלית
24.....	b. Polyphase-Filterbank Interpolation
24.....	4. בקר אינטרפולציה
24.....	a. Modulo -1 Counter Interpolation Control
26.....	b. Recursive Interpolation Control
27.....	פרק 3 – שאלות הכנה
28.....	פרק 4 – מהלך הניסוי
28.....	חלק א' - מפגש ראשון
31.....	חלק ב' - מפגש שני
34.....	נספחים
34.....	א' – תאור תוכן תיקיית Lab Files for students
34.....	ב' – EVM
37.....	ג' – DPLL
37.....	1. חוג PLL אנלוגי
41.....	2. חוג PLL דיגיטלי (DPLL)
44.....	מקורות

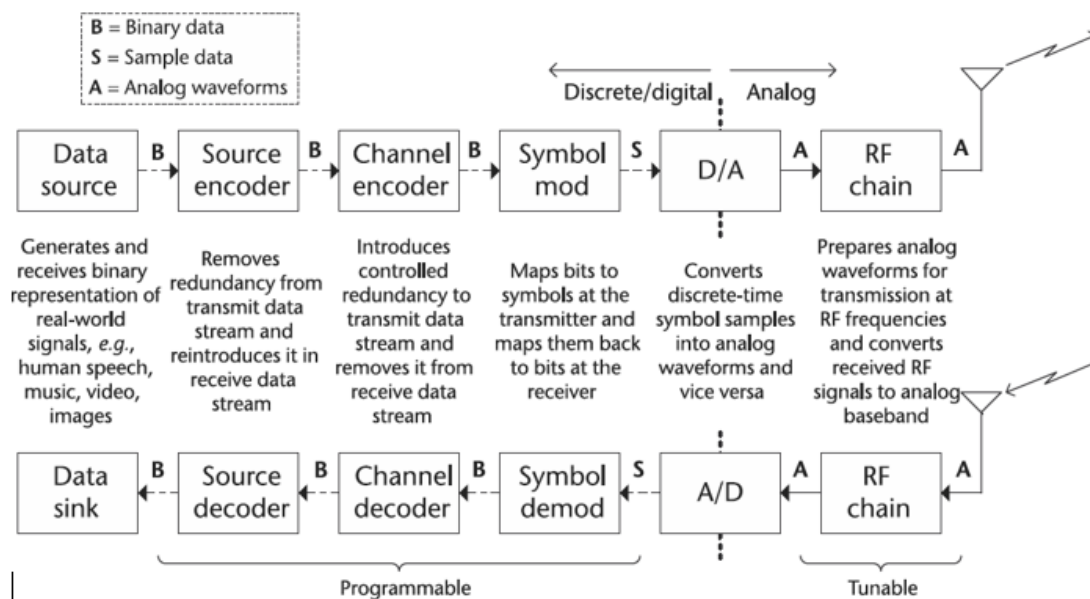


תמצית הניסוי

בעשורים האחרונים התפתחה מאוד הטכנולוגיה של מקמ"ש (מקלט/משדר) הממומש ברובו על ידי רכיבים דיגיטליים הנשלטים על ידי תוכנה ייעודית (דוגמה ידועה לכולנו: טלפון סלולרי).

לטכנולוגיה הזאת קוראים Software Defined Radio (SDR).

דיאגרמת בלוקים מקובלת של SDR מתוארת באיור 1:



איור 1 – דיאגרמת בלוקים של SDR

הבלוקים המסומנים "Programmable" (שידור וקליטה בהתאמה) הם אלה שממומשים בעזרת תוכנה או FPGA. הבלוקים הנ"ל יכולים לעבוד בזמן אמת (לדוגמה טלפון סלולרי) או בקצב איטי בהרבה (לדוגמה קליטת תמונה מלוויין צילום). בכל מקרה האלגוריתמים שמתבססים עליהם דומים והכרתם מאפשר להבין את פעולתן של כל המערכות.

בניסוי הנוכחי נתרכז באלגוריתמים של חלק מהבלוקים Symbol mod (בשידור) ו-Symbol demod (בקליטה). הניסוי יציג את האלגוריתמים השונים ללא התייחסות ספציפית לצורת המימוש, שיכולה להיות FPGA או תוכנה כגון C++, Python וכו'.

האלגוריתמים יוצגו בניסוי בעזרת MATLAB ולכן רצוי ידע בסיסי ב-MATLAB כדי להבין את האלגוריתמים ולהריץ את הניסוי.

ידע תאורטי מקדים הדרוש להבנת חומר הניסוי הוא הכרה של שיטות אפנון דיגיטליות והבנת המושג Error Vector Magnitude (EVM) שהוא מדד חשוב של טיב האפנון.

מי שאינו זוכר או מכיר **חייב** לעין במקורות הקצרים הבאים:

- לקרוא את Digital Modulation in Communication Systems - An Introduction, הנמצא באתר הניסוי, עד עמוד 30 כולל.
- נספח ב' EVM.



כפי שנאמר קודם, האלגוריתמים השונים מומשו לצורך תרגילי המעבדה בעזרת MATLAB Scripts אשר נמצאים בתיקיה בשם Lab Files for students על מחשב הניסוי.

תאור ה-Scripts וסדר הרצתם מופיעים בנספח א'. כל ה-scripts עובדים ומדגימים את האלגוריתמים השונים, אבל הסטודנטים כחלק מניסוי המעבדה יצטרכו להחליף שורות קוד מוסתרות (שורות קוד בעלות סיומת ק.) בשורות קוד גלויות של MATLAB.



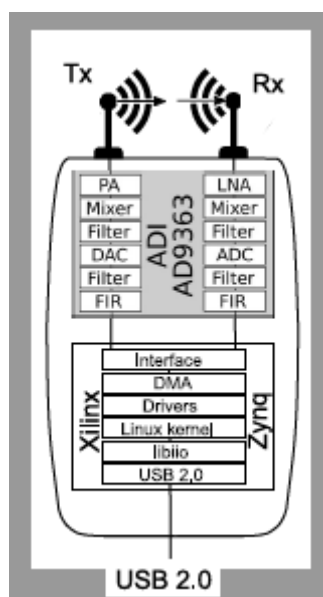
פרק 1 – ציוד הניסוי

החומרה שבעזרתה נבצע את הניסוי מבוססת על מקמ"ש (transceiver) של חברת Analog Devices (ADI) בשם ADALM-PLUTO אשר משווק ע"י החברה בעיקר למטרה לימודית.



איור 2 – SDR מסוג ADALM-PLUTO

היחידה כוללת שני רכיבים עיקריים, כפי שרואים באיור 3:



איור 3 – דיאגרמת בלוקים של PLUTO



שני הרכיבים הם:

- מקמ"ש AD9363
- FPGA מסוג Xilinx Zynq System on Chip (SoC) מסדרת 7000.

היחידה מתחברת לעולם החיצון בעזרת שני קונקטורי RF מסוג SMA לצורך שידור וקליטה (Tx/Rx) וקונקטור USB 2.0 לצורך חיבור (לרוב) למחשב ששולט על היחידה ומנתח את הנתונים הנקלטים. היחידה מקבלת את מתח ההפעלה שלה דרך קונקטור ה-USB.

המעוניין להתעמק ב- SDR הזה, מוזמן להסתכל [כאן](#).

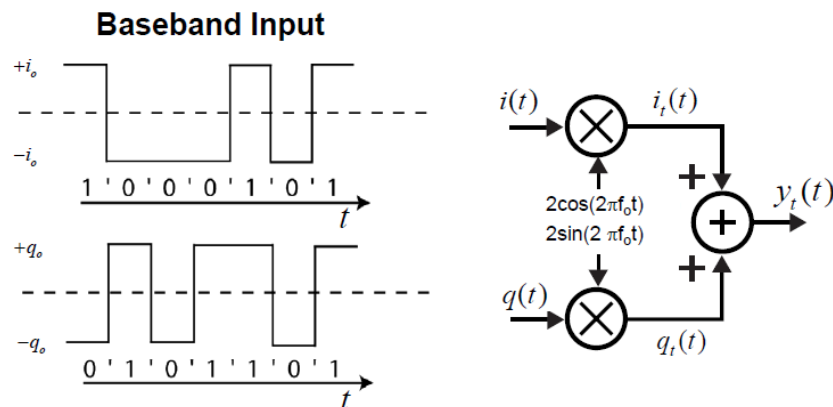


פרק 2 – רקע תאורטי

חלק א' – ערוץ השידור

בערוץ השידור נתרכז בבילוק Symbol mod, אשר ב-PLUTO נמצא בצד ה-RF אחרי ה-DAC.

דיאגרמה מופשטת של בילוק האפנון (שידור) מתואר באיור 4.



איור 4 – בילוק אפנון IQ

לצורך פשטות ההסבר נתייחס לאפנון מסוג QPSK (נהוג לקרוא לו גם 4QAM). כפי שרואים באיור 6, הכניסות i ו- q הם בעלי ערכים דיסקרטיים $\pm i_0$ ו- $\pm q_0$. כתוצאה מכך:

$$i_t(t) = \pm i_0 \cos(2\pi f_o t)$$

$$q_t(t) = \pm q_0 \sin(2\pi f_o t)$$

$$y_t(t) = \sqrt{i_0^2 + q_0^2} \cos(2\pi f_o t + \theta(t))$$

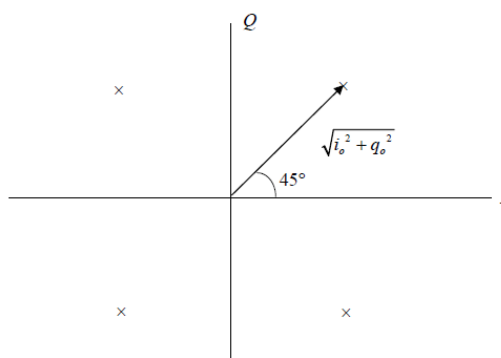
$$\text{where } \theta(t) = \tan^{-1} \frac{\pm q_0}{\pm i_0}$$

$$\text{Given } i_0 = q_0 = 1$$

$$|y_t(t)| = \sqrt{2}$$

$\theta(t)$ can have 4 values

$$45^\circ, 135^\circ, -45^\circ, -135^\circ$$





באופן מעשי מקובל להזין את המאפן בפולסים שעברו סינון מסוג square-root raised cosine כדי להקטין משמעותית את רוחב הסרט של השידור מבלי לפגוע במידע המשודר. בנוסף לכך סינון כזה (כאשר קיים גם בצד הקליטה) מקטין למינימום את תופעת ה- ISI (Inter Symbol Interference) ברגע הדגימה של המידע במקלט.

במקלט צריך להיות סינון זהה מסוג square-root raised cosine כדי שהתגובה המשותפת של שני הפילטרים תהיה מסוג raised cosine. לפרטים נוספים קרא סעיף 4 של Digital Modulation in Communication Systems - An Introduction, הנמצא באתר הניסוי.

חלק ב' – ערוץ הקליטה

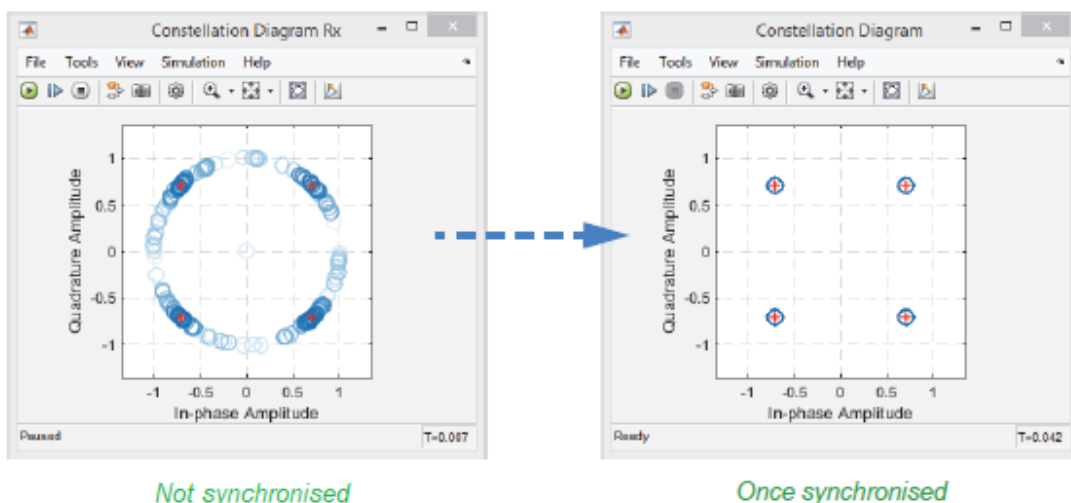
בערוץ הקליטה נתרכז בבחור Symbol demod, שנמצא בצד הספרתי של ערוץ הקליטה ובמקלט SDR בתוך FPGA או ישירות בתוכנה. כתוצאה מכך הוא לא חלק של PLUTO. ב-PLUTO קיימים הרכיבים העיקריים הבאים:

- גלאי IQ בתדר ה-RF המבצע פעולה הפוכה לאפן ה-IQ בצד השידור.
- ממירי ADC מסיגנל אנלוגי לתחום הספרתי.
- מסנן מסוג square-root raised cosine בדומה לצד השידור (שלא חייבים להפעיל אותו במידה ומעוניינים לבצע את הסינון בהמשך השרשרת).

ה-Symbol demod הוא אחד החלקים המעניינים ומאתגרים של מקלט SDR. הוא צריך להתגבר על הבעיות המעשיות שקימות בצמד של משדר/מקלט בשטח:

א. תדר הקליטה (RF) אינו זהה במדויק לתדר השידור (RF).

הסיבה היא שהתדרים האלה מיוצרים ע"י מתנדים הניזונים ממתנדי גביש מדויקים. עד כמה שלא יהיו המתנדים מדויקים, קיים ביניהם הפרש תדר קטן שנמדד ב-ppm (parts per million). לדוגמה אם ההבדל בין הגבישים הוא 1 ppm (שזה הבדל קטן ביותר במערכות מעשיות) אזי בתדר של 1 GHz ההפרש בין תדר השידור לקליטה הוא 1 KHz. כתוצאה מכך הקונסטלציה של האות הנקלט "מסתובבת" ולא נוכל לגלות את האות ששודר.



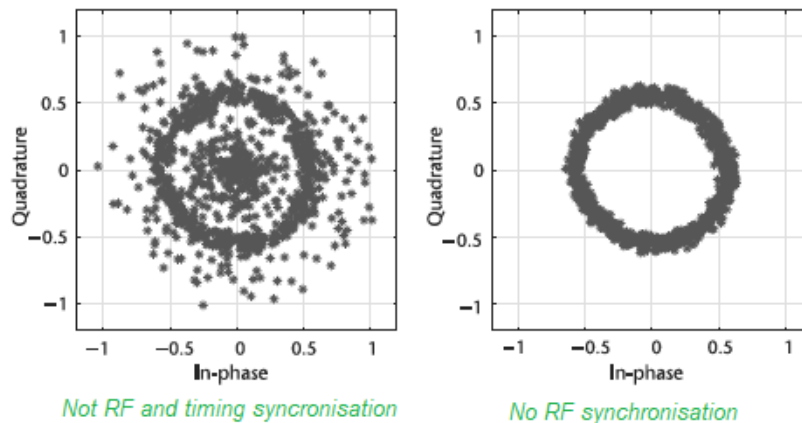
איור 5 – הדגמה לצורך סנכרון של תדר הקליטה לתדר השידור



לכן חיוני לסנכרן בין תדר השידור לתדר הקליטה.

ב. תדר הדגימה של הסיגנל במקלט אינו זהה לתדר הדגימה במשדר.

הסיבה לאי זהות בין תדרי הדגימה היא זהה למקרה הקודם (תדר הדגימה נגזר ממתנד הגבישי של המשדר או המקלט בהתאמה). לכן יש לבצע סנכרון גם בין תדרי הדגימה. איור 5 מתאר את נושא סנכרון של תדר הקליטה לתדר השידור, אך לצורך פשטות התאור מניח שתדרי הדגימה מסונכרנים. באופן מעשי תדרי הדגימה אינם מסונכרנים ולכן בקליטה האות אינו נדגם ברגע אופטימלי וכתוצאה מכך ה-ISI גבוה.

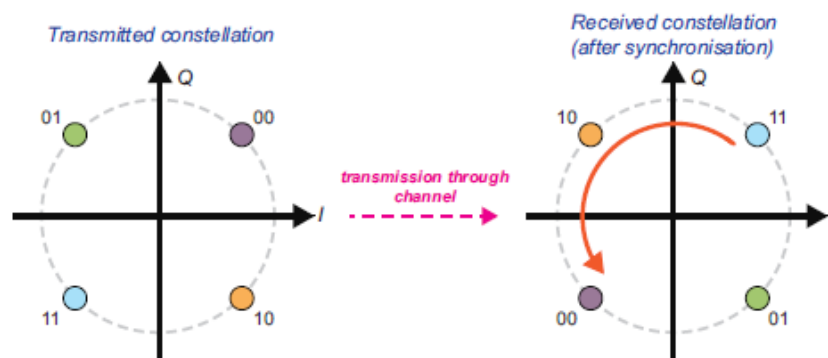


איור 6 – הדגימה לצורך סנכרון של תדרי הדגימה

מהאיור הנ"ל רואים בבירור שבהתחלה הדגימות לא אופטימליות ולכן ה-ISI גבוה. לאחר סנכרון תדרי הדגימה, נושא ה-ISI מסתדר אבל עדין יש צורך לסנכרן את תדר הקליטה לתדר השידור (כפי שנעשה באיור 5) כדי שנקודות הקונסטלציה לא תסתובבנה.

ג. אי ודאות של פאזת הקונסטלציה

גם אחרי סנכרון תדר, קיימת אי ודאות בפרוש האות הנקלט בגלל שכתוצאה מהסנכרון הקונסטלציה יכולה להתייבב על מספר הזזות פאזה בהתאם לסוג האפנון (לדוגמה במקרה של QPSK יש 4 אפשרויות הנבדלות אחת מהשנייה ב-90 מעלות).



איור 7 – דוגמה של סיבוב קונסטלציה QPSK ב-180 מעלות לאחר סנכרון תדר

באיור 7 רואים שכתוצאה של סיבוב הקונסטלציה הסימבול "00" ששודר מפורש במקלט כ-"11". זהו נושא שיש לטפל בו, אך לא נגע בנושא זה בניסוי הנוכחי.



ד. סנכרון חבילה (frame or packet)

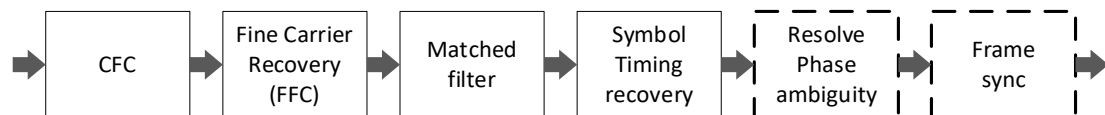
באופן מעשי המידע משודר בחבילות במבנה כללי שדומה למוצג באיור 8.



איור 8 – דוגמה של מבנה של חבילה (frame)

יש צורך חיוני לזהות את תחילת החבילה כדי להיות מסוגלים לקרוא את המידע ששודר. גם נושא זה לא יכולה בניסוי הנוכחי.

קטע שרשרת הקליטה שנטפל בניסוי מורכב מהבלוקים שבאיור הבא:



איור 9 – הבלוקים של שרשרת הקליטה

הבלוקים מסודרים לפי הסדר המקובל של עיבוד האות הנקלט, למרות שהרבה פעמים תהיו עדים לכך שמשנים במקצת את סדר הבלוקים כדי לקבל ביצועים אופטימליים, בהתאם לסוג האפנון הנקלט. הבלוקים הם:

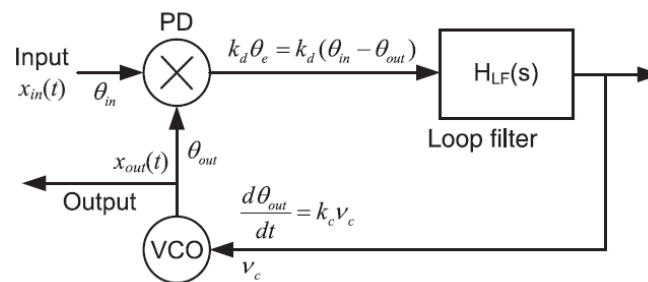
1. CFC (Coarse Frequency Offset Correction). זהו תיקון תדר קליטה ראשוני (בצורה "גסה") כדי להקל על תיקון תדר הדגימה ותדר הקליטה העדין בהמשך.
2. בלוק של שיחזור עדין של תדר הקליטה (FFC).
3. מסננת מתואמת (מסוג square-root raised cosine).
4. שיחזור תדר הדגימה של הסימבול.
5. פתרון אי ודאות פאזה.
6. סנכרון חבילה.

המשך ההסברים יתרכזו בכל אחד מהבלוקים.

חלק ג' – PLL דיגיטלי

לפני שנדון בבלוקים השונים של ערוץ הקליטה, נסקור בקצרה את התאוריה שמאחורי PLL דיגיטלי שהוא חלק מרכזי להבנת רוב האלגוריתמים שנדון בהם בניסוי הנוכחי.

עד שנות ה-70 של המאה הקודמת רוב התקשורת הייתה מבוססת על טכנולוגיה אנלוגית וגם חוגי ה-PLL היו אנלוגיים (APLL).



איור 10 – דיאגרמת בלוקים מופשטת של APLL

ה-APLL ביסודו חוג בקרה שהכניסה שלו היא הפאזה של $x_{in}(t)$ והמוצאה היא הפאזה של $x_{out}(t)$. המרכיבים הבסיסיים של החוג הם:

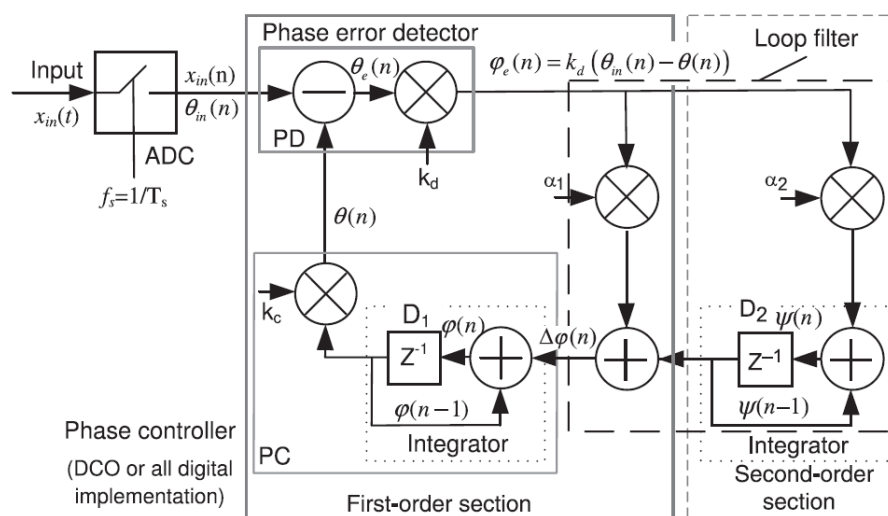
- גלאי פאזה PD, שתפקידו להפיק את שיחסי להפרש הפאזה בין שתי האותות שהוזכרו קודם.
- מסנן החוג (Loop filter) שקובע את הדינמיקה של החוג.
- מתנד מבוקר מתח (VCO) שמפיק את הפאזה של האות במוצא.

פונקציית התמסורת של החוג נתונה ע"י:

$$H_{APLL}(s) = \frac{\Theta_{out}(s)}{\Theta_{in}(s)} = \frac{k_c k_d H_{LF}(s)}{s + k_c k_d H_{LF}(s)}$$

היות שברוב השימושים מעוניינים שהפאזה של אות המוצא תעקוב אחרי הפאזה של אות הכניסה, החוג הנפוץ ביותר הוא חוג מסדר שני (ז"א שבנוסף לקוטב אחד שתורם ה-VCO קיים עוד קוטב נוסף של $H_{LF}(s)$). חוג אנלוגי מסוג זה נחקר לעומק במאה שעברה ולכן עד היום נהוג להשוות ביצועים של חוג סיפרתי לתוצאות ידועות של חוג אנלוגי.

תוך כדי התבססות הטכנולוגיה הספרתית מרכיבים שונים של חוג ה-PLL הפכו להיות ספרתיים (DPLL) והיום נהוג להשתמש בחוגים שממומשים באופן מלא בתוכנה. כמוכן היום נהוג לפעמים לדלג על לימוד של חוג APLL ומתרכזים ישר בניתוח ותכנון חוג DPLL (זה גם דורש שליטה טובה בהמרת z).



איור 11 – דיאגרמת בלוקים של חוג DPLL מסדר שני

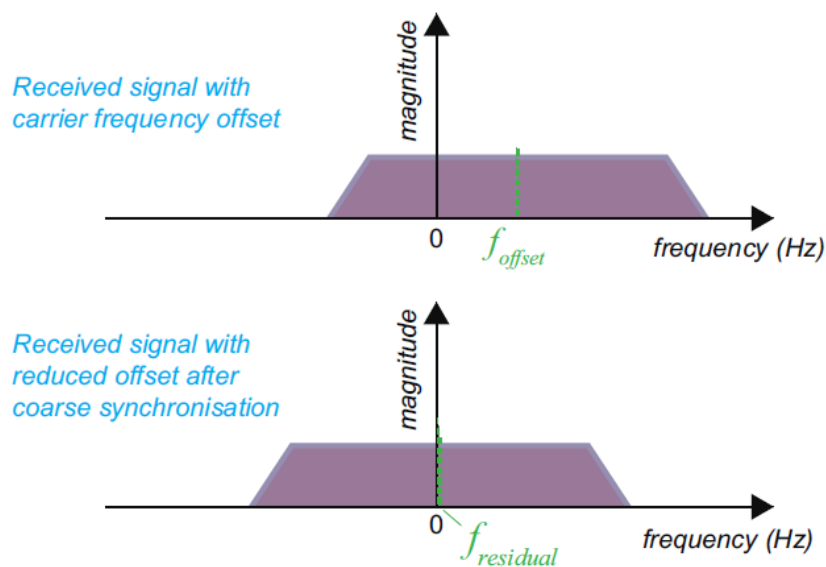


המרכיבים הבסיסים של החוג הם:

- גלאי פאזה PD שתפקידו כמו בחוג האנלוגי להפיק אות שיחסי להפרש הפאזות, אך הפעם זה מבוצע בתוכנה. תראו בהמשך שזה הרכיב המאתגר ביותר בחוג וקיימות הרבה שיטות מימוש.
 - המסנן של החוג שמורכב משני ענפיים מקביליים (ענף הגבר רגיל וענף שכולל אינטגרטור).
 - PC (Phase controller) שהוא גם כן ממומש בעזרת אינטגרטור.
- להבנה מעמיקה יותר של ה-DPLL ונוסחאות התיכון יש לעין בנספח ג.

חלק ד' – CFC

הבעיה שאלגוריתם ה-CFC מנסה לפתור מודגמת באיור הבא:

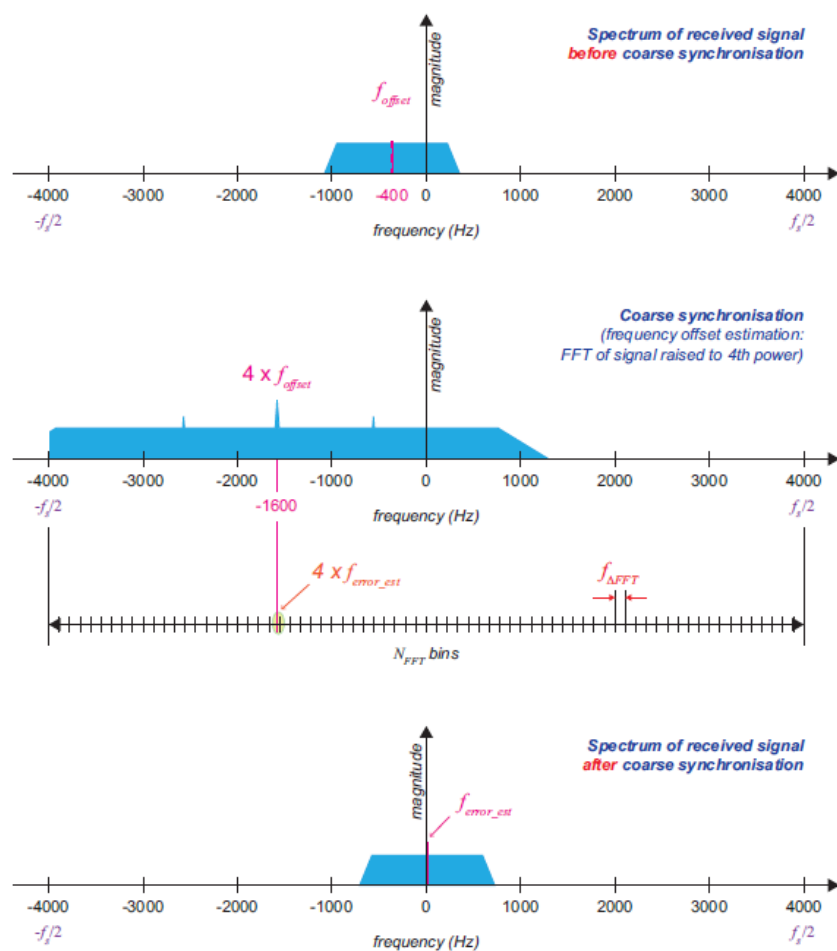


איור 12 – פעולת CFC

קיימים הרבה אלגוריתמים כדי לבצע CFC ואנו נציג אחד מהפשוטים והנפוצים ביותר. לצורך הפעלת האלגוריתם חייבים לדעת את סוג אפנון הסיגנל ובעיקר את אינדקס אפנון הפאזה M (עבור M QPSK שווה ל-4). שלבי האלגוריתם:

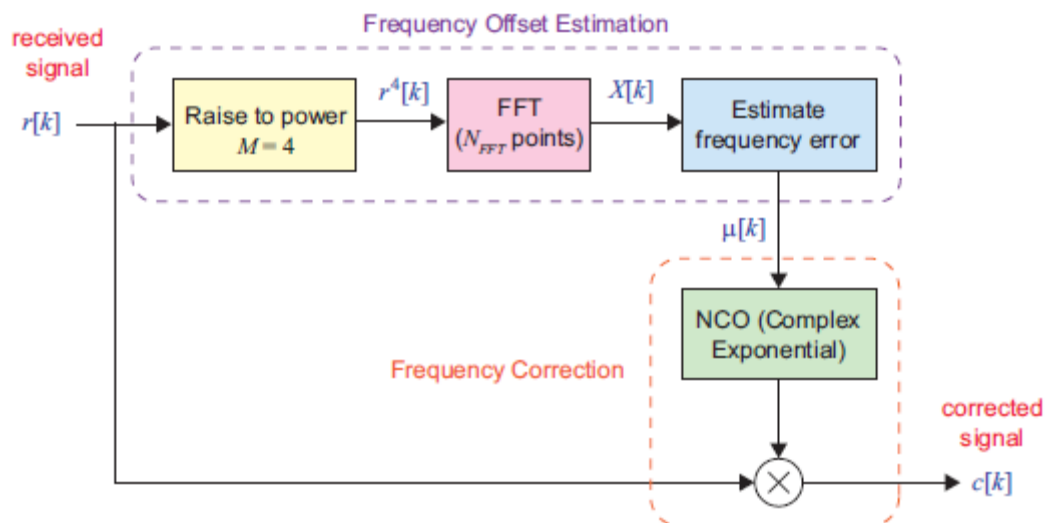
- מעלים את הסיגנל לחזקת M וכתוצאה מכך מקבלים טון חזק בתדר שהוא פי M מתדר ה-offset.
- מקבלים את ה-offset האמיתי ע"י חלוקת ערך ה-offset לאחר העלה בחזקה ב-M.

אפשר לממש את האלגוריתם ע"י חישוב ה-FFT של האות (לאחר חזקה M) וזיהוי ה-bin בעל העוצמה הגבוהה ביותר. השיטה מודגמת באיור הבא עבור $f_s = 8000$ Hz (תדר דגימה) והזנת תדר של האות הנקלט של 400 Hz.



איור 13 – אלגוריתם ה-CFC

דיאגרמת הבלוקים של אלגוריתם ה-CFC:



איור 14 – דיאגרמת בלוקים של אלגוריתם ה-CFC



יש לשים לב שה-CFC אינו מאפס באופן מוחלט את שגיאת התדר שבין המשדר למקלט, אלא מקטין את השגיאה לערך נמוך המאפשר לאלגוריתם FFC לבצע תיקון מדויק (ה-FFC מבוסס על חוג DPLL בעל רוחב סרט נתון).

במידה שתדר הדגימה של האות הוא f_s ומספר נקודות ה-FFT הוא N_{FFT} אז הרזולוציה של ה-bins נתונה ע"י:

$$f_{\Delta FFT} = \frac{f_s}{N_{FFT}}$$

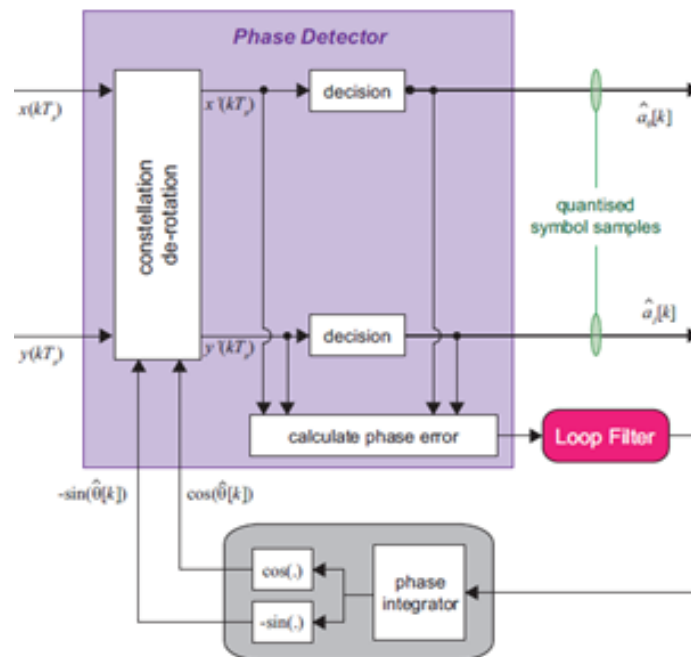
לכן נקבל שגיאה מקסימלית של שיערוך שגיאת התדר:

$$f_{error_est} = \frac{f_{\Delta FFT}}{2M}$$

חלק ה' – FFC

לאחר ביצוע שלב ה-CFC עוברים לשלב של תיקון שארית סטית התדר ופאזה.

השיטה המתבקשת לתיקון שארית סטית התדר ופאזה היא ליצר אותות סינוס וקוסינוס באותו תדר ופאזה של האות בכניסת בלוק ה-FFC ולהכפיל אותו באות הכניסה כדי לגלות את פאזת האותות I ו-Q ב-BB. אפשר לבצע את הפעולה הזאת במגוון שיטות, כאשר אחת השיטות המקובלות היא לבצע אותה ישירות ב-BB (על הרכיבים I ו-Q של האות) כפי שמתואר באיור הבא:



איור 15 דיאגרמת בלוקים של FFC

כפי שרואים, הפעם האלגוריתם מבוסס על חוג משוב (feedback) בניגוד ל-CFC שהוא אלגוריתם מסוג feedforward. דגימות הסימבולים ב-BB הן $(x(kT_s), y(kT_s))$ במישור IQ, כאשר k הוא אינדקס דגימה של הסימבול ו- T_s הוא קצב הסימבול. הנקודה של הקואורדינטות הנ"ל מושוות מבחינת פאזה



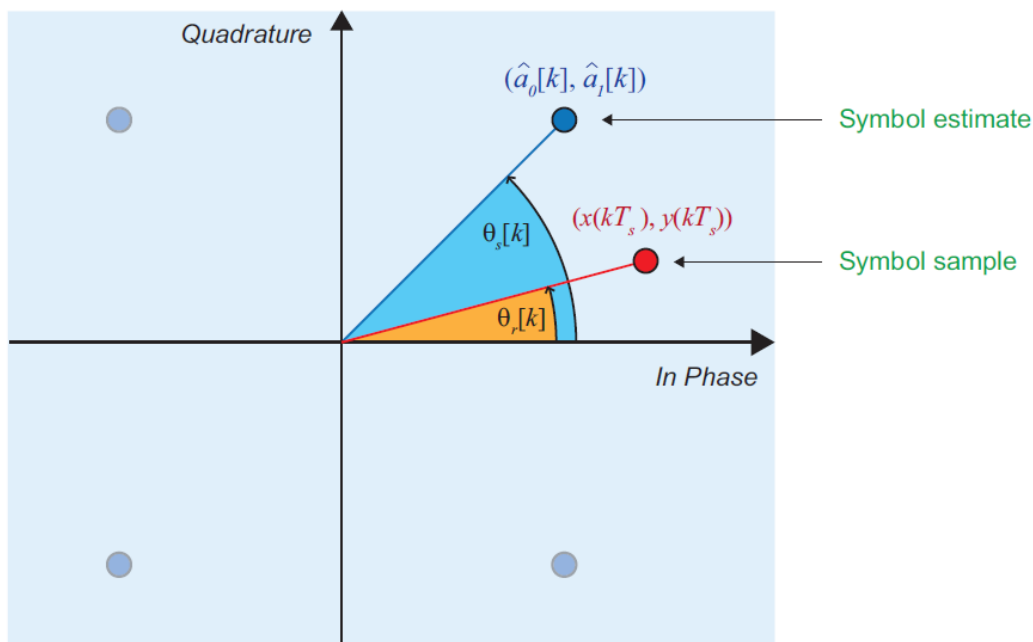
לנקודת יחוס של הקונסטלציה הרצויה (צריכים לדעת מראש איזו קונסטלציה משודרת) כדי לחשב את שגיאת הפאזה.

קיימות שתי גישות בנושא:

- שיטה שנקראת decision directed כאשר אין אינפורמציה בקשר לסימבולים המשודרים. במקרה זה מחליטים על זהות הסימבול הנקלט כסימבול הקרוב ביותר לדגימה שנקלטה.
- שיטה שנקראת data aided כאשר המקלט יודע איזה סימבולים שודרו (שלרוב משודרים ב-preamble) ומחשבים את טעות הפאזה בהתאם.

ההסבר שלנו יתבסס על השיטה הראשונה. זה אומר שאחרי הסנכרון יתכן שקיימת אי ודאות של פאזת הקונסטלציה (עבור QPSK בכפולות של 90°) שיש לתקן בנפרד בהמשך שרשרת הקליטה.

במקרה של QPSK הדגימות ממופות לסימבול הקרוב ביותר כפי שמתואר באיור הבא:



איור 16 חישוב שגיאת הפאזה עבור כל דגימה

בצורה כזאת מיצרים דגימות סימבולים משוערכים $(\hat{a}_0[k], \hat{a}_1[k])$. לדגימות הסימבולים (לפני ואחרי ההחלטה) פאזות $\theta_r[k]$ ו- $\theta_s[k]$ בהתאמה. את הפאזות אפשר לחשב מהנוסחאות:

$\Theta_s = \arctan(\hat{a}_1[k] / \hat{a}_0[k])$ ו- $\Theta_r = \arctan(y(kT_s) / x(kT_s))$. מתוך כך ניתן לחשב את הבדל הפאזות כ:

$$\Theta_e[k] = \theta_r[k] - \theta_s[k]$$

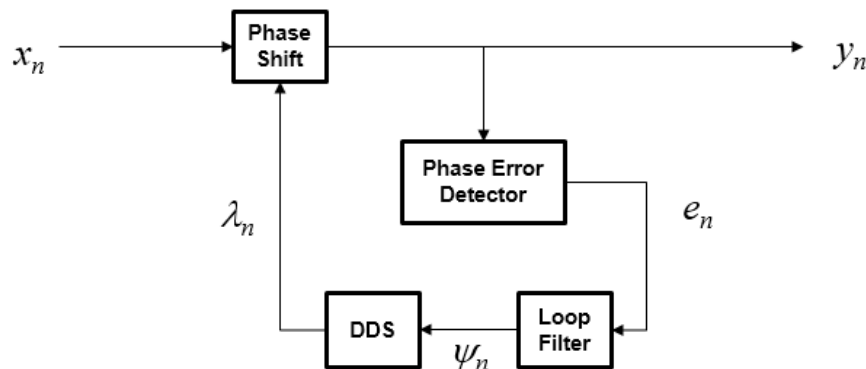
טעות הפאזה הזאת (המחושבת בבלוק calculate phase error) מועברת לחוג משוב המיצר אקספוננציאל קומפלקסי שמתקן בצורה דינמי את מיקום הסימבול (בבלוק constellation de-rotation). עבור QPSK ההחלטות (של הבלוקים decision) הן:

$$\hat{a}_0[k] = A \times \text{sgn}\{x'(kT_s)\}$$

$$\hat{a}_1[k] = A \times \text{sgn}\{y'(kT_s)\}$$



לצורך כתיבת האלגוריתם של FFC נשתמש במונחים של חוג נעול פאזה בדומה למובא בנספח [ג](#). נתייחס לדיאגרמת הבלוקים הבאה כדי לחשב את y_n מתוך x_n לפי הנוסחה $y_n = x_n e^{i\lambda_n}$.



איור 17 חוג ה- FFC

הבלוק Phase Error Detector מחשב את שגיאת הפאזה התלוי בסוג האפנון.

Modulation	Phase Error
QAM or QPSK	$e_n = \text{sgn}(\text{Re}\{x_n\} \times \text{Im}\{x_n\}) - \text{sgn}(\text{Im}\{x_n\} \times \text{Re}\{x_n\})$
BPSK or PAM	$e_n = \text{sgn}(\text{Re}\{x_n\} \times \text{Im}\{x_n\})$

כדי להבטיח קיום חוג יציב שגיאת הפאזה מעוברת דרך מסנן חוג מסוג biquadratic:

$$\psi_n = g_1 e_n + \psi_{n-1}$$

כאשר ψ_n היא המוצא של הפילטר עבור דגימה n ו- g_1 הוא הגבר האינטגרטור. הגבר האינטגרטור נתון ע"י:

$$g_1 = 4(\theta^2/d)/K_p K_0$$

כאשר θ ו- d נתונים ע"י:

$$d = 1 + 2\zeta\theta + \theta^2 \quad \theta = B_n/(\zeta + 1/4\zeta)$$

B_n הוא רוחב הסרט לרעש של החוג מנורמל לתדר הסימבולים ו- ζ הוא מקדם הריסון. K_0 הוא קבוע של ה- DDS ו- K_p (הגבר של גלאי שגיאת הפאזה) תלוי בסוג האפנון. עבור BPSK, PAM, QAM, QPSK, $K_p=2$.

המוצא של הפילטר ψ_n מועבר לכניסת מסנן נוסף (DDS) שהוא גם מסנן מסוג biquadratic:

$$\lambda_n = (g_p e_n + \psi_{n-1}) + \lambda_{n-1} \quad \text{כאשר:}$$

$$g_p = 4\zeta(\theta/d)/K_p K_0$$

תחום הנעילה המנורמל (לתדר הסימבול) הנתון ב- rad/sec הוא:

$$(\Delta f)_{\text{pull-in}} \approx \min(1, 2\pi\sqrt{2}\zeta B_n)$$

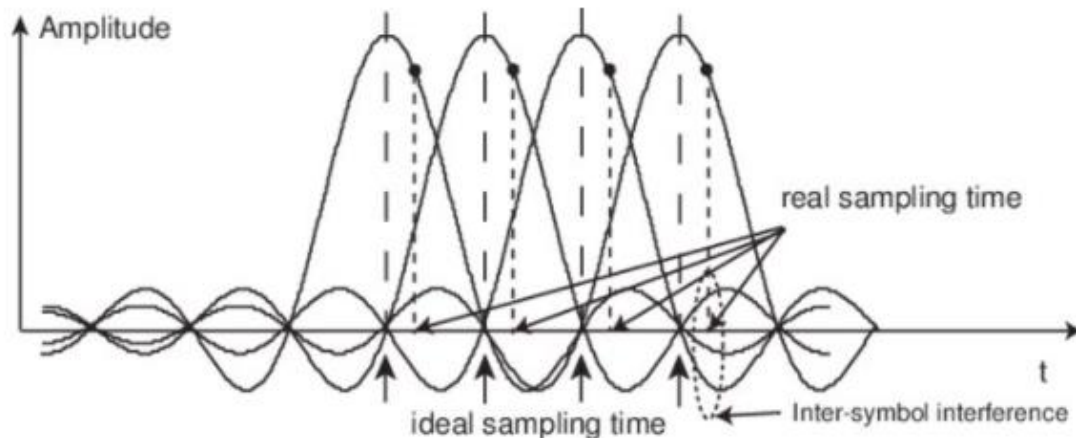
זמני נעילת התדר ופאזה (הנתונים כמספר דגימות) משוערכים ע"י:

$$T_{\text{FL}} \approx 4 \frac{(\Delta f)_{\text{pull-in}}^2}{B_n^3} \quad T_{\text{PL}} \approx \frac{1.3}{B_n}$$



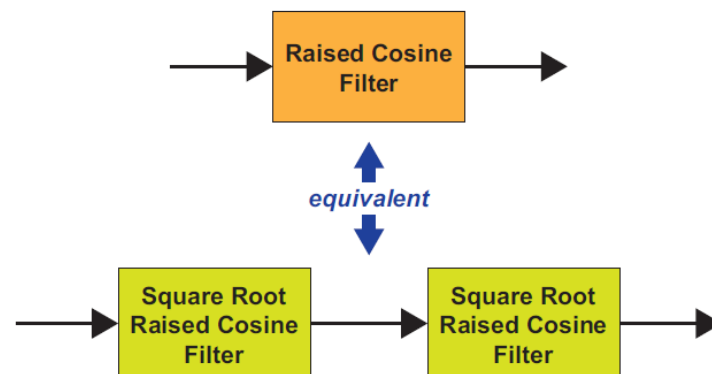
חלק ו' – Timing Recovery

כפי שלומדים בקורסי התקשורת, השיטה היעילה להגבלת הספקטרום של המידע המשודר מצד אחד וביטול הפרעות בין הביטים (ISI) המשודרים מצד שני היא השימוש במסנן Raised Cosine (RC).



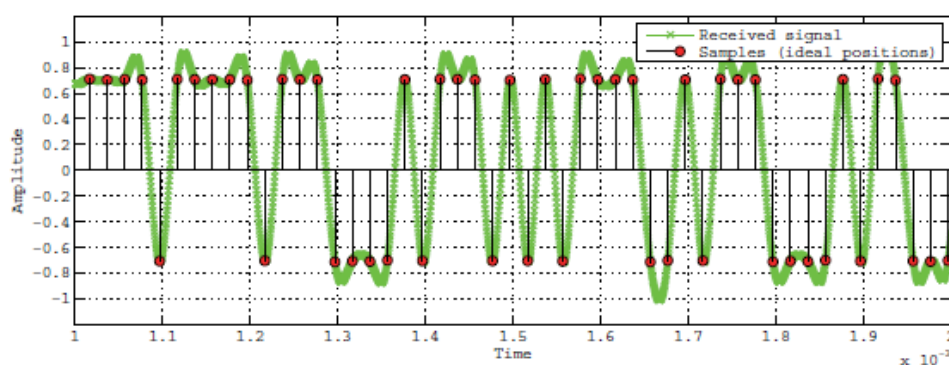
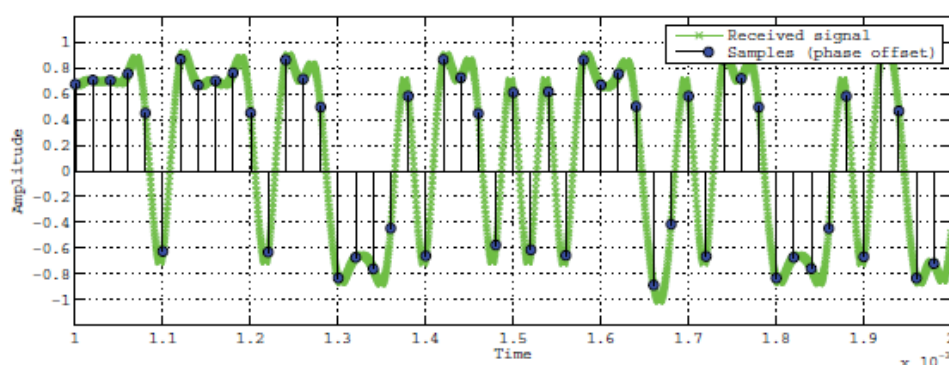
איור 18 שימוש במסנן RC ודגימה ברגע הנכון לצורך ביטול ISI

לרוב פונקציית המסנן מחולקת בין השידור לקליטה, כאשר בכל צד קיים מסנן מסוג Root Raised Cosine (RRC). אומנם מסנן בודד מסוג RRC אינו גורם לאפס ISI, שני מסננים ביחד כן מביאים לתוצאה הרצויה.



איור 19 שילוב של שני מסננים SQRC לקבלת מסנן RC

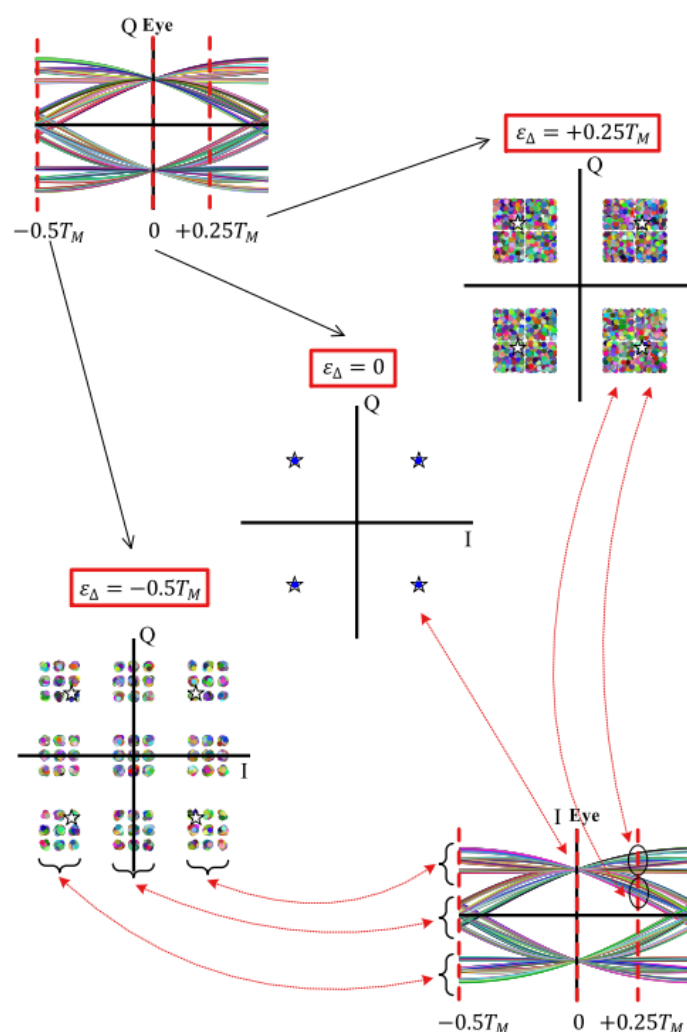
כדי שהמערכת תתפקד בצורה נכונה הדגימה של האות צריכה להתבצע בתיזמון נכון. ללא פעולה יזומה הדגימה בצד המקלט אינה מתבצעת ברגע האופטימלי בגלל שתדר ופאזה של אות הדגימה אינם תואמים את הנתונים בצד המשדר. את התוצאה ההרסנית ניתן לראות בחלק העליון של האיור הבא:



איור 20 השוואה בין סיגנל נקלט ונדגם בתזמון לא נכון לדגימת הסיגנל לאחר תיקון תזמון הדגימה



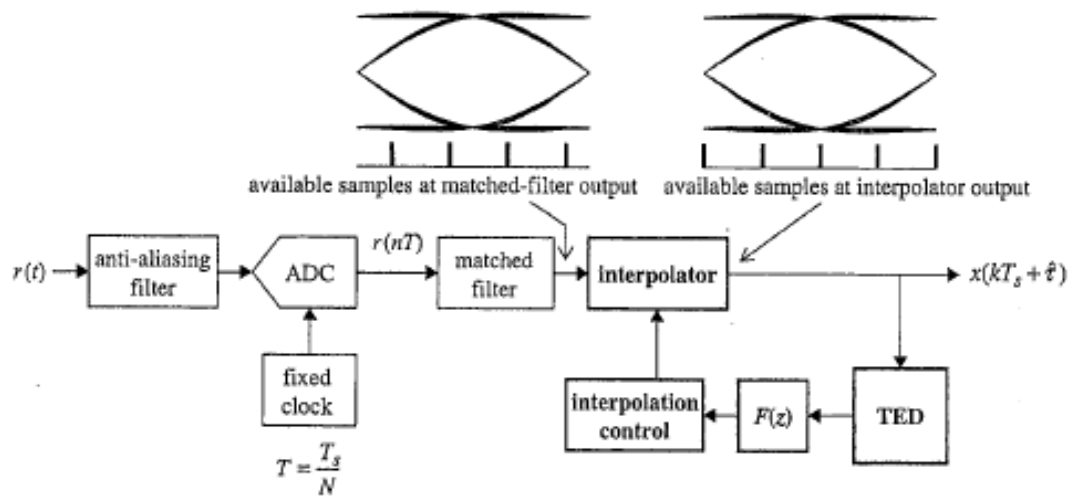
האיור הבא מראה איך דיאגרמת הקונסטלציה מתעוותת עבור שגיאות תזמון שונות (עבור אפנון QPSK). שימו לב שקל לנתח את התופעה במידה ובוחנים את דיאגרמות העין:



איור 21 דוגמאות של דיאגרמות קונסטלציה המתקבלות עבור תזמוני דגימה שונים



לצורך שחזר התדר (ופאזה) של שעון הסימבולים משתמשים לרוב בחוג PLL כפי שמתואר באיור הבא:



איור 22 חוג PLL לשחזור זמן הדגימה האופטימלי

חוג ה-PLL מורכב מהבלוקים הבאים:

- גלאי שגיאת תיזמון (Timing Error Detector (TED)).
- מסנן חוג $F(z)$ (Loop Filter).
- בקר אינטרפולציה (Interpolation Controller).
- אינטרפולטור.

לרוב גם המסננת המתואמת של ערוץ הקליטה היא חלק של מעגל שיחזור התיזמון. אפשר לממש כל אחד מהבלוקים במגוון שיטות (ראה מקור [2]). באופן מעשי כמעט לכל אחד מהבלוקים קיים קצב דגימה שהוא אופטימלי לקבלת התוצאה הרצויה. היות שהקצביים אינם זהים עבור הבלוקים השונים, כשמחברים את הבלוקים ביחד יש לדאוג להתאמת קצבי הדגימה. זהו נושא שימושי ונרחב בעיבוד ספרתי של אותות ונקרא Multirate Signal Processing. ראה מקור [4].

1. TED

באופן כללי ה-TED מיצר אות שגיאה פעם אחת בסימבול בהתבסס על שיערוך התיזמון הנוכחי. לכן השגיאה הדיסקרטית מתעדכנת בקצב הסימבול. אות השגיאה המיוצר ע"י ה-TED והתלוי בשגיאת התיזמון דומה מבחינת תפקוד לגלאי פאזה של PLL שמיצר אות שהיא פונקציה של שגיאת פאזה.

בדומה ל-FCC גם לגלאי שגיאת תיזמון יש שתי שיטות מימוש:

- שיטה שנקראת decision directed כאשר אין אינפורמציה בקשר לסימבולים המשודרים. במקרה זה מחליטים על זהות הסימבול הנקלט כסימבול הקרוב ביותר לדגימה שנקלטה.
- שיטה שנקראת data aided כאשר המקלט יודע איזה סימבולים שודרו (שלרוב משודרים ב-preamble) ומחשב את טעות הפאזה בהתאם.

כדי לא לסבך את ההסברים, אנו נצמד לשיטה הראשונה.

קיימים מימושים רבים של TED. אנו נעבור בקצרה על שתיים מהם.



a. גלאי שגיאת תזמון מסוג Maximum Likelihood Timing Error Detector (MLTED)

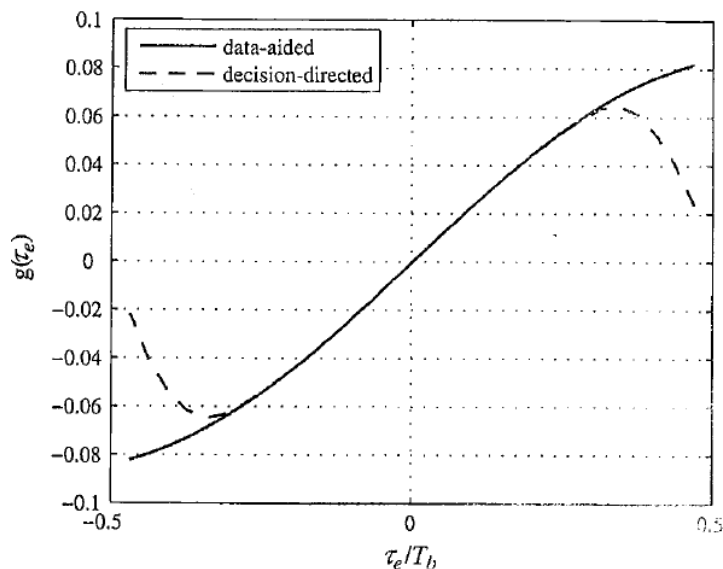
זהו גלאי ה-TED הבסיסי שעבורו קיים פיתוח מתמטי מדויק.

גלאי מסוג זה מפיק את אות שגיאת התזמון מתוך הנגזרת של דיאגרמת העין של האות (הכולל גם תיקון סימן). המטרה היא לאתר את רגע התזמון כאשר הנגזרת אפס שמתאים לתזמון האופטימלי. נוסחת עקומת הגלאי נתונה ע"י:

$$\begin{aligned} g(\tau_e) &= E \left\{ a(k) \frac{d}{dt} x(kT_s + \tau) \right\} \\ &= E \left\{ a(k) \frac{d}{dt} K \sum_m a(m) r_p((k-m)T_s - \tau_e) \right\} \\ &= KE_{avg} \dot{r}_p(-\tau_e) \end{aligned}$$

הנוסחה משתמשת בנגזרת כתלות בזמן של פונקציית האוטוקורלציה של צורת הפולס בנקודה $-\tau_e$. לעקומה הגלאי נהוג לקרוא עקומת S בגלל הצורה הגרפית שלה.

עבור MLTED עקומת ה-S היא:



איור 23 עקומת S של גלאי שגיאת תזמון מסוג MLTED

ההגבר של הגלאי K_p הוא השיפוע של העקומה הנ"ל בנקודה $g(0)$.

ל-TED מסוג זה מספר חסרונות כגון תלות האות המופק בהספק הנקלט ויצור רעש פנימי שקשור לעצם שיטת העבודה של הגלאי. על החיסרון הראשון נהוג להתגבר ע"י שימוש במעגל AGC (Automatic Gain Control).

יש לזכור גם שה-MLTED הינו לרוב מערכת מסוג multirate. באלגוריתמים שאנו נטפל אות הכניסה ל-MLTED הינו דגימה אחת לסימבול ובמוצא מתקבלות N דגימות לסימבול (כמספר הדגימות לסימבול של האות הנקלט).



b. גלאי שגיאת תיזמון מסוג (ELTED) Early-Late Timing Error Detector

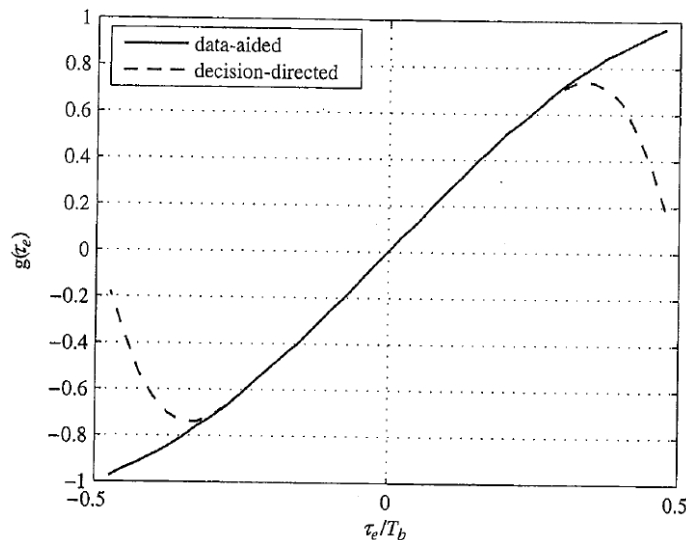
הגלאי עובד לרוב בקצב סיגנל כניסה של 2 דגימות לסימבול ובודק את ההפרש בין שני ערכי דגימה $\frac{1}{2}$ סימבול לפני ואחרי זמן השגיאה המשוער.

$$e(k) = a(k) [x((k + 1/2)T_s + \hat{\tau}) - x((k - 1/2)T_s + \hat{\tau})]$$

לאחר פיתוח "קצר" אפשר לראות שנוסחת עקומת ה-S נתונה במקרה זה ע"י:

$$g(\tau_e) = KE_{avg} [r_p(T_s/2 - \tau_e) - r_p(-T_s/2 - \tau_e)]$$

והעקומה היא:



איור 24 עקומת S של גלאי שגיאת תזמון מסוג ELTED

ההגבר K_p של ELTED הוא השיפוע בנקודה $g(0)$ והוא יחסי לעוצמת הסיגנל הנקלט K והאנרגיה הממוצעת של סימבול E_{avg} .

גם הפעם יש להשתמש במעגל AGC כדי לשמור על עוצמה קבועה של סיגנל בכניסת הגלאי. עבור גלאי מסוג זה בעיה של רעש פנימי קצת יותר פחותה מאשר עבור MLTED.

c. גלאים נוספים

בנוסף לשני הגלאים שהסברנו, קיימים עוד גלאים שלכל אחד יתרונות וחסרונות משלהם.

- Zero-Crossing Timing Error Detector (ZCTED)
- Gardner Timing Error Detector (GTED)
- Mueller and Müller Timing Error Detector (MMTED)

המעוניינים ללמוד על הגלאים הנ"ל מתבקשים לעיין במקור [2].

2. מסנן החוג $F(z)$

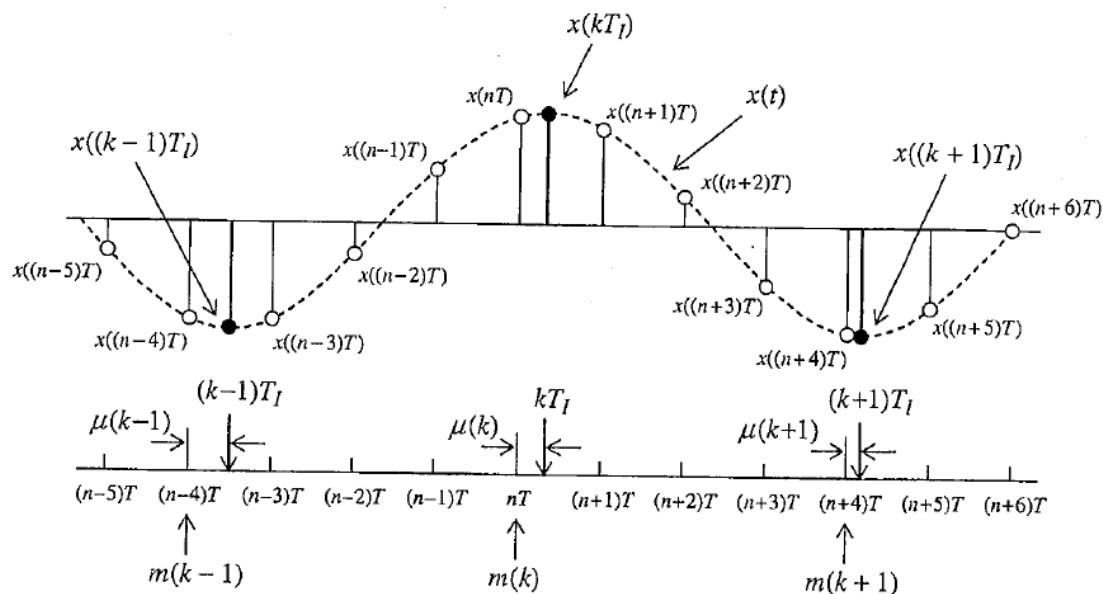
מסנן החוג הוא לרוב מסוג PI. הסבר ונוסחאות של מסנן מסוג זה נמצאים בנספח ג'.



3. אינטרפולטור

תפקיד האינטרפולטור הוא להזיז את נקודת הדגימה לנקודה האופטימלית על פי מידע שמתקבל מבקר האינטרפולציה. באיור הבא ניתן לראות אות בכניסת האינטרפולטור שנדגם 4 פעמים לסימבול כאשר אף דגימה אינה בנקודה האופטימלית. מטרת האינטרפולטור היא ליצר דגימה $x(kT_I)$ ברגע kT_I הנקראת the k -th interpolant. הפעולה הזאת נקראת interpolation. כאשר ה- k -th interpolant נמצא בין הדגימות $x(nT)$ ו- $x((n+1)T)$ האינדקס n נקרא k -th basepoint index ומסומן כ- $m(k)$. הזמן kT_I גדול מ- $m(k)T$ בערך שקטן מ- T (הזמן שבין הדגימות). היחס בין הערך הזה ל- T מסומן כ- $\mu(k)$ ונקרא the k -th fractional interval. $0 \leq \mu(k) < 1$ ומוגדר בעזרת: $\mu(k)T = kT_I - m(k)T$.

האינטרפולטור חייב לקבל את מידע של ה- basepoint index ו- fractional interval כדי להפיק את ה- interpolant. כפי שנראה המידע מתקבל מבקר האינטרפולטור.



איור 25 הדגימות להבנת פעולת האינטרפולטור

פילטר האינטרפולציה האידיאלי הוא מסוג IIR. זה פילטר מסובך למימוש, במיוחד שה- fractional interval משתנה. נהוג להשתמש בפילטר מסוג FIR שהוא קירוב של הפילטר האידיאלי.

שיטת אינטרפולציה נוספת היא להשתמש ב- polyphase-filterbank interpolator.

a. אינטרפולציה פולינומיאלית

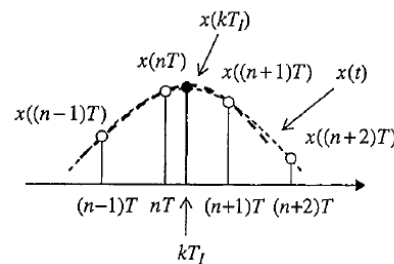
ניתן לתאר בקירוב את צורת הגל של סימבול בעזרת פולינום מסוג:

$$x(t) \approx c_p t^p + c_{p-1} t^{p-1} + \dots + c_1 t + c_0.$$

ברגע שהמקדמים ידועים, אפשר לחשב את $x(kT_I)$:

$$x(kT_I) \approx c_p (kT_I)^p + c_{p-1} (kT_I)^{p-1} + \dots + c_1 (kT_I) + c_0.$$

הפולינום המקובל ביותר הוא פולינום ריבועי ($p=2$). הפולינום הזה נותן תוצאה מדויקת וקלה למימוש. תאור קירוב בעזרת פולינום ריבועי:

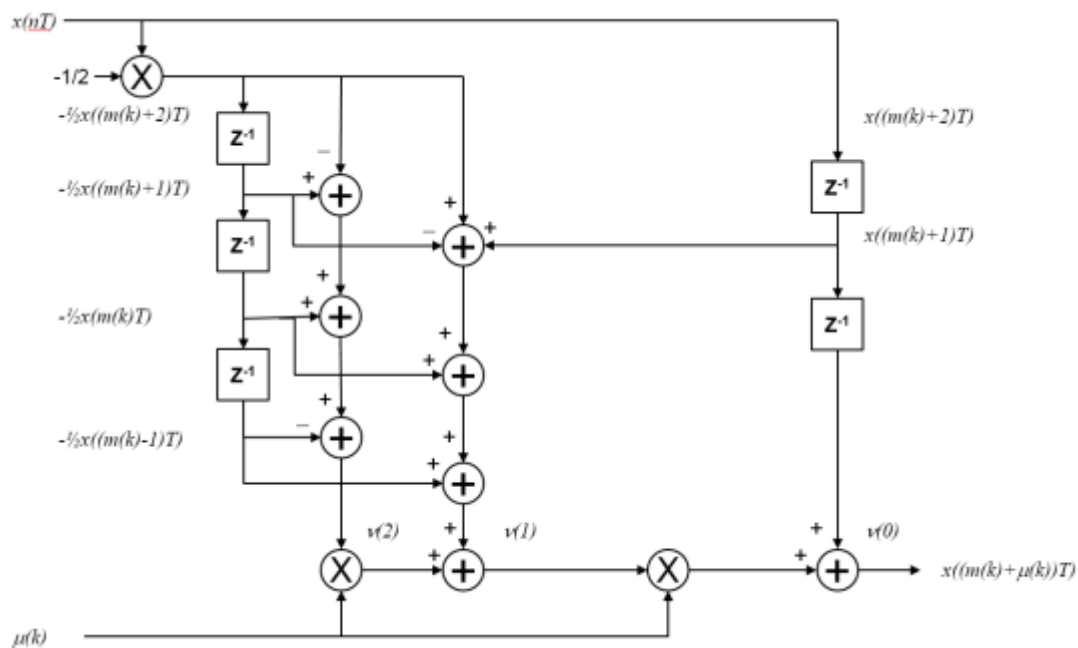


$$x(t) \approx c_2 t^2 + c_1 t + c_0$$

$$x(kT_I) \approx c_2 (kT_I)^2 + c_1 (kT_I) + c_0$$

איור 26 מציאת מיקום הדגימה עבור אינטרפולטור ריבועי

מימוש אינטרפולציה פולינומאלית ריבועית (אינטרפולטור Farrow):



איור 27 מימוש אינטרפולטור ריבועי מסוג Farrow

b. Polyphase-Filterbank Interpolation

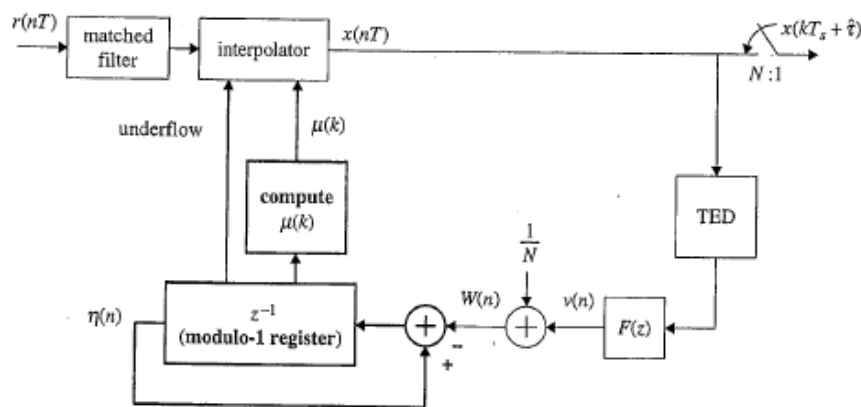
זאת שיטת אינטרפולציה נוספת מקובלת מאוד. המעוניין ללמוד על כך יכול לעיין בכל ספר הדן ב-Multi Rate Digital Signal Processing ובמיוחד במקור [2].

4. בקר אינטרפולציה

תפקיד בקר אינטרפולציה הוא לספק לאיטרפולטור את הפרמטרים $m(k)$ ו- $\mu(k)$. באופן מעשי לא מצינים את ה- $m(k)$ בצורה מפורשת אלא מסמנים אותו ע"י סיגנל בשם strobe. תפקיד הבקר במימוש הנוכחי דומה לתפקיד ה-DDS בנספח ג'. קיימות שתי שיטות עיקריות למימוש הבקר, כפי שנציג בהמשך.

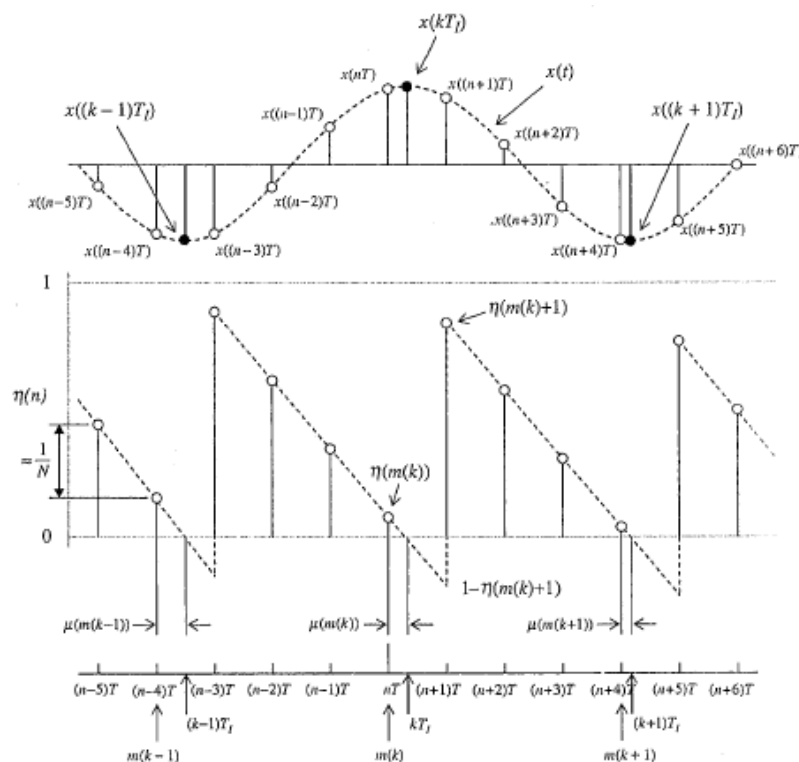
a. Modulo -1 Counter Interpolation Control

הרעיון של השיטה הזאת הוא להשתמש במונה כלפי מטה מסוג modulo -1 שמתאפס פעם ב- N דגימות ובמצב נעול ההתאפסות ויצירת פולס ה-strobe תואם את $m(k)$.



איור 28 חוג תזמון דגימה עם Modulo-1 Counter

המונה סופר כלפי מטה בקפיצות ממוצעות של $1/N$ כך שאיפוס המונה והפקת פולס strobe מתרחשים בממוצע פעם ב- N דגימות. המוצא של מסנן החוג $\gamma(n)$ שולט על הערך המדויק של הספירה כלפי מטה. בפעולה תקינה איפוס המונה ויצירת ה- strobe מתרחשים שעון אחד אחרי $m(k)$ כפי שמתואר באיור הבא:



איור 29 האותות בחוג המשתמש ב- Modulo-1 Counter

באופן כללי, ערך המונה מקיים את נוסחת הרקורסיה:

$$\eta(n+1) = (\eta(n) - W(n)) \bmod 1$$

כאשר $W(n) = 1/N + \gamma(n)$ היא הכניסה למונה והיא השיערוך של היחס T_s/T .

לאחר התחשבות ב- modulo-1 מקבלים:



$$\eta(m(k)+1) = 1 + \eta(m(k)) - W(m(k))$$

היות שערכי המונה $\eta(m(k))$ ו- $1 - \eta(m(k)+1)$ יוצרים משולשים דומים, מקבלים:

$$\frac{\mu(m(k))}{\eta(m(k))} = \frac{1 - \mu(m(k))}{1 - \eta(m(k)+1)}$$

מכאן מקבלים:

$$\mu(m(k)) = \frac{\eta(m(k))}{1 - \eta(m(k)+1) + \eta(m(k))} = \frac{\eta(m(k))}{W(m(k))}$$

מחזור האיפוס (בדגימות) של מונה היורד modulo -1 הוא:

$$\begin{aligned} \frac{1}{W(n)} &= \frac{1}{\frac{1}{N} + v(n)} \\ &= \frac{N}{1 + Nv(n)} \end{aligned}$$

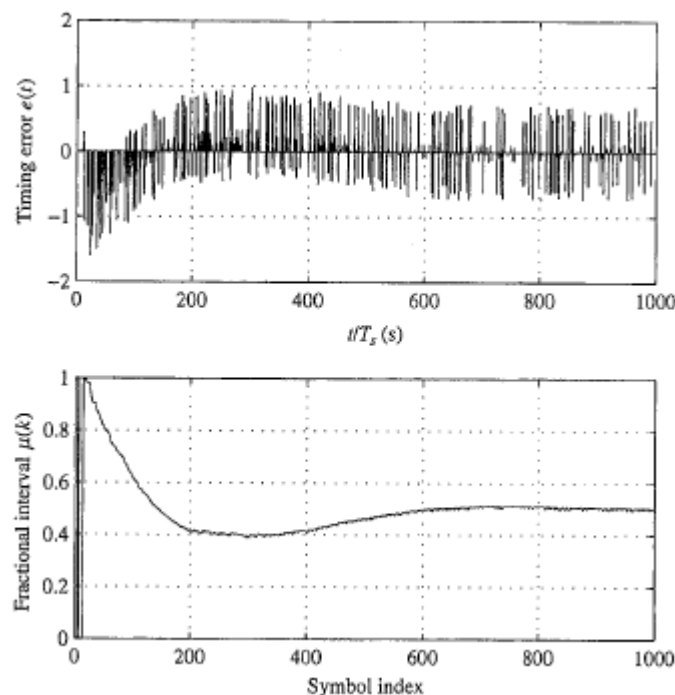
כאשר החוג נעול, $v(n)$ הוא אפס בממוצע ומחזור האיפוס של המונה הוא N דגימות בממוצע.

שימו לב שבגלל שהמונה הוא מונה יורד ההגבר שלו הוא $K_0 = -1$.

b. Recursive Interpolation Control

זאת עוד שיטה (פחות פופולרית). אפשר ללמוד על השיטה הזאת ממקור [2].

דוגמה של גרפים של שגיאת תיזמון ($e(t)$) במוצא ה-TED) ו- $\mu(k)$ תוך כדי נעילת החוג נתונה באיור הבא עבור חוג בעל גלאי שגיאת תיזמון מסוג ELTED:



איור 30 דוגמה של שגיאת תזמון ו- $\mu(k)$ של חוג תיזמון דגימה



פרק 3 – שאלות הכנה

יש לשמור את התשובות לשאלות ההכנה ב-LabAdmin לפני המפגש הראשון. את התשובות יש להגיש מודפסות בפורמט PDF. תשובות בכתב יד לא תתקבלנה.

1. מעבירים מידע באפנון 16QAM ב-bit rate של 24Mbps. מה הוא ה-symbol rate?
2. תציין לפחות יתרון אחד וחסרון אחד של אפנון 256QPSK.
3. עבור אפנון מסוים נתון $\text{symbol rate} = 10\text{MBs (Mega Baund/sec)}$ (קצב symbols נמדד ב-baunds). חשב את α כאשר ה-occupied bandwidth הנמדד הינו 15MHz.
4. עבור אפנון QPSK ארבעת נקודות הקונסטלציה צפויות להקלט במישור IQ בנקודות (מתח) $(1,1)$, $(-1,1)$, $(-1,-1)$ ו- $(1,-1)$. במקלט הקונסטלציה נקלטת מסובבת 10° בכיוון השעון. מה הוא ה-EVM ב-% וב-dB?
5. מתוך הגדרת EVM תוכיח ש: $\text{EVM(dB)} = -\text{SNR(dB)}$
6. עבור חוג PLL תמצאו את הנוסחה המקשרת בין B_n ל- f_{3dB} ומתוך כך את היחס בין B_n ל- f_{3dB} . כתלות במקדם הריסון ζ . סרטטו גרף של היחס הנ"ל עבור ζ בתחום 0.2 – 3. נוסחה כזאת עוזרת לתכנן את הקבועים של החוג לפי נתון של f_{3dB} שהוא לרוב מקובל יותר מאשר השימוש ב- B_n .

המלצה: לצורך מוכנות, רצוי לקרוא את מהלך הניסויים לפני הגעה למעבדה.



פרק 4 – מהלך הניסוי

במהלך הניסוי מוצגות שאלות מודגשות שיש לענות עליהם בדו"ח של המפגש. בין ההרצות השונות מומלץ למחוק את הגרפים שהתקבלו ולנקות את החלונות בעזרת הפקודות `clear all` ו-`clc`.

חלק א' - מפגש ראשון

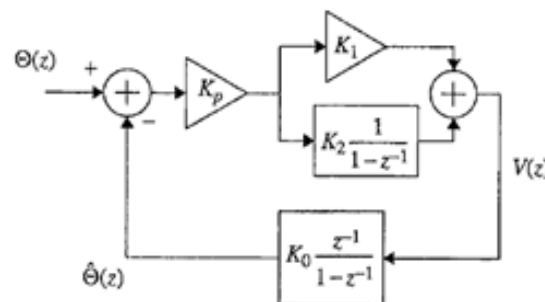
במפגש הראשון נלמד על פעולת חוג מסוג DPLL ונתרגל את האלגוריתמים השונים של סנכרון תדר הקליטה לתדר השידור ותיזמון דגימה אופטימלית של האות הספרתי בעזרת אות סינטי שניתן לשלוט בכל הפרמטרים שלו.

כפי שנאמר, בשלב ראשון נתרגל פעולת חוג DPLL.

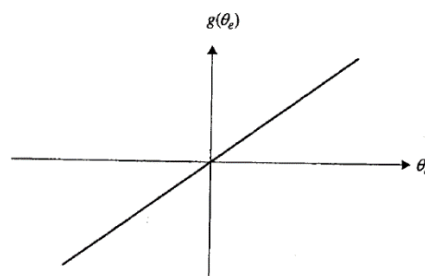
תפתחו ב-MATLAB את הקובץ `Digital_Phase_Locked_Loop.m` ותקראו אותו במטרה להבין את מבנהו. הקובץ מחולק לשני חלקים עיקריים:

- ניתוח DPLL בתחום התדר.

עבור ניתוח בתחום התדר נשתמש במודל לינארי של החוג בהתאם לאיור ג-3 (b)



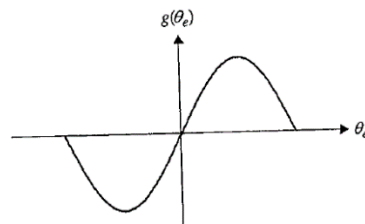
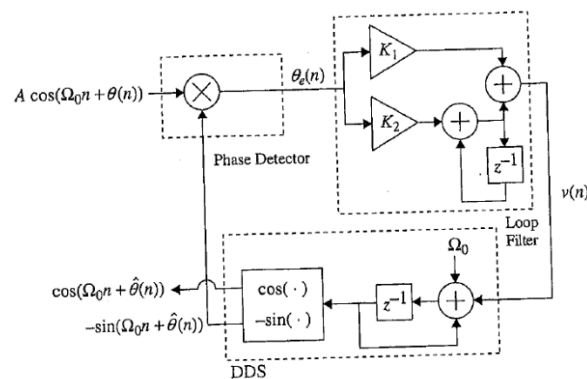
עבור המקרה הזה גלאי הפאזה הוא לינארי:



בסימולציה נניח שקבוע הגלאי הוא $K_p=1$.

- ניתוח DPLL בתחום הזמן.

במקרה הזה נשתמש במודל הבא:



במקרה הזה גלאי הפאזה ממומש ע"י מכפלה של אות הכניסה לחוג באות במוצא ה- DDS (מודול פאזה של 90 מעלות).

1. באיזה שורה של קוד ה- MATLAB מופיעה נוסחת גלאי הפאזה? מה הערך המקסימלי של $g(\theta_e)$ הגלאי?

תריצו כרגע את הקובץ ותתבוננו על המסך. תוך זמן קצר יופיעו הגרפים של החישובים כאשר הגרף האחרון הוא גרף דינמי המדגים את נעילת החוג בתחום הזמן.

נסתכל על הגרפים לפי הסדר:

- גרף 1 מתאר את שגיאת הפאזה (ההבדל בין פאזת אות הכניסה לפאזה של אות היציאה) כתלות בזמן של המודל עם גלאי הפאזה הלינארי. ברגע הראשון השגיאה היא π ועם הזמן השגיאה מתאפסת. היות שגלאי הפאזה לינארי, במקרה הזה שגיאת הפאזה היא גם אות השגיאה ביציאת גלאי הפאזה.
- גרף 2 מתאר את תגובת התדר של המודל הלינארי.
- גרף 3 ו- גרף 4 מציגים את שגיאות החוג ביציאת גלאי הפאזה והבדל הפאזה בין פאזת אות הכניסה לפאזה של אות היציאה. שימו לב שבגלל שגלאי הפאזה אינו לינארי שני הגרפים אינם זהים.
- גרף 5 מציג את סיגנלים בכניסת החוג ובמוצא ב- DDS כתלות בזמן. הגרף הזה מתקשר לגרף הדינמי שראיתם מיד לאחר ההרצה.

2. תסבירו, בהתבסס על אופיין גלאי הפאזה של החוג הלא לינארי, את ההבדל בין גרף 3 לגרף 4 (תבססו את ההסבר על צורת אופיין הגלאי).

3. על פי תוצאות הסימולציה, איזה חוג מבצע נעילה מהר יותר (למרות שממומשים עם אותם פרמטרים) ולמה?

4. בשורה 33 של הקוד קיימות שתי שורות קוד מוסתרות המחשבות את הקבועים a_1 ו- a_2 המופיעים בנוסחת תגובת התדר של DPLL. תציינו בדו"ח את שתי השורות הנ"ל.

5. בשורה 10 של הקוד נתון רוחב הסרט לרעש, המנורמל לתדר הדגימה (במקרה שלנו רוחב הסרט הוא 20KHz). שנו את הערך הנתון (0.001) לערך שעבורו נקבל רוחב סרט 3dB



של 20KHz. תריצו את הקובץ ותצרפו את גרף 2 שמתקבל לדו"ח (ביחד עם החישוב של הערך החדש).

- כרגע נמשיך ונבחן את האלגוריתמים השונים של נעילת תדר וסנכרון תיזמון הסימבולים בעזרת אות סנטתי המדמה שידור אמיתי, אך ניתן לשלות בכל הפרמטרים שלו.
- תפתחו את הקובץ Test_Signal.m ותעיינו בו. בתחילת הקובץ קיימים כל הפרמטרים של הסיגנל ובהמשך הפקודות המממשות את הסיגנל ומצירות דיאגרמת קונסטלציה.

6. תצינו בדו"ח את הפרמטרים העיקרים של הסיגנל, כגון: מספר הדגימות לסימבול, סוג האפנון, קצב הדגימה, שגיאת התדר הנקלט ביחס לתדר המקלט, היחס של אות לרעש (במצב הראשוני מדובר באופן מעשי באות נקי ללא רעש).

- תריצו את הקובץ ותתבוננו על דיאגרמת הקונסטלציה המתקבלת. בדיאגרמה מוצגים 10000 הסימבולים האחרונים של האות המשודר. אל תמחקו את הגרפים ומשטחי העבודה עד שלא נכתב במפורש.
- תריצו כרגע את הקובץ CFC.m. הקובץ ממומש בדיוק לפי המוסבר [בחלק ד'](#) של הרקע התאורטי. תעברו על הקובץ ותנסו להבין את תוכנו.

7. בשורה 20 של הקובץ קיימת שורה מוסתרת שמחשבת את הסיגנל המתוקן y_{CFC} בעזרת שגיאת התדר $CFOffset$ (בפרקי זמן t) המופעל על הסיגנל הנקלט y . תרשמו בדו"ח את הנוסחה.

- הגרף הראשון (Figure 1) מציג את ספקטרום הסיגנל הנקלט לאחר העלה בחזקה רביעית. תמקמו את הסמן על השיא של האות המרכזי ותקראו את הזזת התדר של האות שהעלה בחזקה רביעית.

8. מתוך כך חשבו את הזזת התדר של האות ששודר. האם זה תואם בקרוב רב את הנתון של האות ששודר?

- תסתכלו על הגרף השני (Figure 2) שמציג את הספקטרום בחזקה רביעית של הסיגנל המתוקן. תוודאו שהפעם הזזת התדר הרבה יותר קטנה מאשר בגרף הראשון.
- תעיינו בקונסטלציה המתקבלת הפעם. כמו בקונסטלציה הקודמת גם הפעם מוצגים 10000 הסימבולים האחרונים של האות המשודר.
- תעיינו כרגע בקובץ Fine_carrier_sync.m. הקובץ ממומש בדיוק לפי המוסבר [בחלק ה'](#) של הרקע התאורטי. תעברו על הקובץ ותנסו להבין את תוכנו. בראש הקובץ ישנם מספר פרמטרים של החוג ובהמשך האלגוריתם ודיאגרמת הקונסטלציה.
- תריצו את הקובץ. גרף מספר שלוש (Figure 3) מציג את פאזת האות במוצא ה-DDS (שהנגזרת שלה הוא תיקון התדר) ובגרף הקונסטלציה רואים את התמונה המתקבלת לאחר תיקון עדין של התדר.

9. תבחנו את שלושת דיאגרמות הקונסטלציה שהתקבלו עד עכשיו. תסבירו דיאגרמה אחרי דיאגרמה מה רואים (רמז: עבור ההסבר של הדיאגרמה האחרונה תעזרו גם באיור 23).

- כרגע תעברו לקובץ symbol_synchronizer.m. הקובץ ממומש בדיוק לפי המוסבר [בחלק ו'](#) של הרקע התאורטי. ניתן לבחור מתוך אחד ה- MLTED או ELTED. האינטרפולטור הוא מסוג Farrow ריבועי. תבחנו את הקובץ כדי להבין את מבנהו והקשר שלו לפונקציות משניות.
- תריצו את הקובץ (עם MLTED). מופיעות שתי תצוגות דינמיות. באחת רואים את התכנסות הדגימות לקונסטלציה QPSK, בתצוגה השנייה רואים במקביל את שינוי μ כתלות ב- k . לאחר מכן מופיעה דיאגרמת קונסטלציה של 10000 הסימבולים האחרונים ודיאגרמת העיין המתאימה (תזכרו שהאלגוריתם מפיץ במוצא דגימה אחת לסימבול). בנוסף באיור 4 רואים



- את התיזמון (רועש בגלל הרעש העצמי של ה-TED אשר מסונן ע"י החוג), באיור 5 את מוצא הבקר PI (שימו לב שהרעש העצמי סונן בצורה חזקה) ובאיור 6 את שגיאת התיזמון (μ).
- בשלב זה אפשר למחוק את כל הגרפים ולנקות את משטחי העבודה. כרגע נריץ מספר ניסויים:
- נבדוק את פעולת החוג כאשר גם קצב הדגימה במקלט שונה מאשר במשדר. לצורך כך נפעיל את שורה 13 בקובץ Test_Signal.m ונחסום את שורה 14 (בעזרת %). כרגע תריצו זה אחר זה בסדר הנכון את ארבעת קבצי האלגוריתם.

10. תעינו באיור 6. רואים שלמרות שהחוג התכנס לקונסטלציה QPSK השגיאה μ לא התייצבה על ערך קבוע. תסבירו את הסיבה.

- אפשר למחוק את כל הגרפים ולנקות את משטחי העבודה. בקובץ Test_Signal.m תשנו את ה-SNR מ-60 dB ל-15 dB. תריצו שוב את ארבעת הקבצים בסדר הנכון.

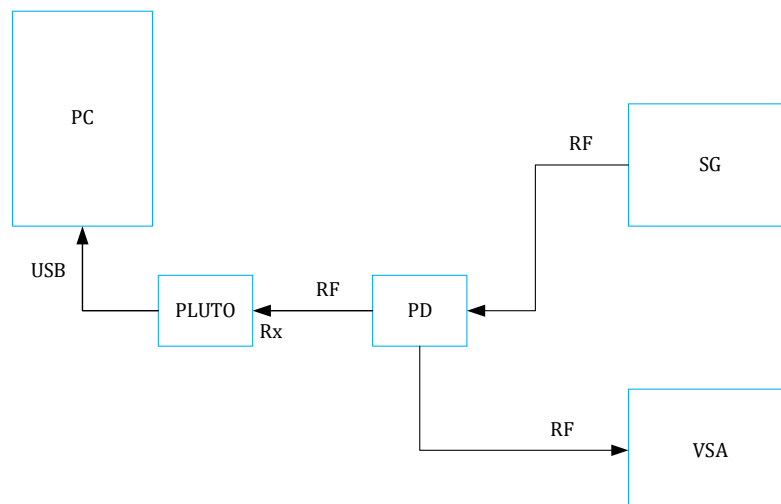
11. תסתכלו על דיאגרמת העיין ועל הקונסטלציה הסופית שהם רועשים בגלל הסיגנל הרועש שבחרנו. תקראו מתוך דיאגרמת הקונסטלציה את Avg EVM (dB). האם תואם את הציפיות? תסבירו.

חלק ב' - מפגש שני

במפגש השני נלמד במעבדה בעזרת מחולל אותות (Signal Generator) איך ליצר סיגנל מאופנן QPSK בהתאם למפרט נתון בהתבסס על אפליקציית MATLAB.

את סיגנל ה-QPSK נחבר למחבר הקליטה (Rx) של SDR מסוג PLUTO ונעביר את ה-BB של האות לצורך עיבוד אל תוכנת ה-MATLAB. לאחר מכן נפעיל על האות את האלגוריתמים השונים של הסינכרוניים ונבדוק שמתקבלת התוצאה הרצויה.

- בתחילת הניסוי תקבלו מערך מחובר בהתאם לאיור הבא:



- בשלב ראשון יש לוודא שהמחשב מזהה את PLUTO. לצורך כך מריצים את הקובץ Tx_Init.m במידה שהמחשב הצליח לזהות את PLUTO התוכנה מציגה את נתוני ברירת מחדל של הרכיב.
- נתכנת את ה-Signal Generator בעזרת אפליקציית MATLAB. נא ללחוץ על לשונית APPS בחלק העליון של חלון MATLAB ומהשורה העליונה לבחור את האפליקציה Wireless Waveform Generator ומתוכו את האופציה של סיגנל QAM.
- תמלאו את נתוני הסיגנל בהתאם לטבלה הבאה:



▼ QAM Waveform Configuration

Modulation order: 4

Symbol mapping: Gray

Bit source: PN15

Number of input bits: 300000

Output symbol rate (sym/s): 10e+6

▼ Filtering Configuration

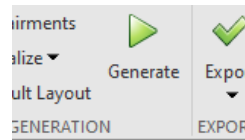
Filtering: Root raised co...

Oversampling factor: 4

Rolloff factor: 0.5

Filter span (symbols): 20

- לאחר מילוי הנתונים יש ללחוץ על Generate



- בצד ימין של המסך יופיעו הסיגנל (BB), הספקטרום של הסיגנל ודיאגרמת הקונסטלציה. תבדקו שהספקטרום והקונסטלציה תואמים את הציפיות בהתאם לפרמטרים של הסיגנל שהאזנתם קודם.
- כרגע תלחצו על לשונית TRANSMITTER בחלק העליון של האפליקציה. תזהה את ה-Signal Generator וכל מה שנישאר הוא למלא את הפרטים הבאים:

▼ Signal Conditioning & Transmission

Center frequency (GHz): 1.8

Output power (dBm): -10

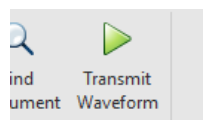
Baseband sample rate (Hz): 40000000

Interpolation factor: 1

Output sample rate (Hz): 40000000

☐ Tukey windowing

- השלב הבא הוא ללחוץ על Transmit Waveform



- האפליקציה תטען את ה-Signal Generator והסיגנל יועבר ל-PLUTO ול-VSA.
- בשלב הזה יש לקרוא למדריך כדי שידריך אותכם בהפעלת ה-VSA.
- תבדקו את צורת הספקטרום, הקונסטלציה וה-EVM שמתקבלים. האם תואמים את הציפיות?
- כרגע תפתחו את הקובץ Capture_Rx_Pluto.m. תפקידו של הקובץ לקלוט את הסיגנל מ-PLUTO. בשורה 16 של הקובץ מגדירים את כמות הדגימות שיש לקלוט (במקרה שלנו 3e5)



וקובעים שם לקובץ עם סיומת bb. כדאי לתת שם קובץ (אפילו ערוך) המתווכו אפשר לזכור איזה סיגנל נשמר.

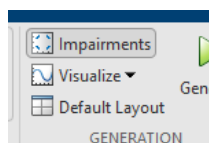
- מריצים את הקובץ ומוודים שהסיגנל נקלט בתיקיה בשם שקבענו.
- בשלב זה אפשר לחזור לאפליקציה ולעצור את השידור.
- כדי להפעיל את אלגוריתם הסנכרון והתיזמון על הקובץ שנשמר יש להריץ את הקבצים בסדר הבא (לפני כן רצוי לסגור את חלון האפליקציה ליצירת הסיגנל):
 - o Read_captured_Rx.m (לא לשכוח לעדכן בשורה 15 את שם הקובץ הרצוי)
 - o CFC.m
 - o Fine_carrier_sync.m
 - o symbol_synchronizer.m (במצב MLTED)

12. תבחנו את דיאגרמות הקונסטלציה בשלבים השונים (תצרפו את הדיאגרמות לדו"ח) ותסבירו מה אתם רואים. תשמרו להמשך את דיאגרמת העיין ואת ערך ה- EVM שקיבלתם.

- תיצרו עוד סיגנל דומה לקודם בהבדל יחיד שהפעם Rolloff factor = 0.2.
- תריצו את כל קבצי האפליקציה כאשר בשורה 16 של symbol_synchronizer.m תשנו את ה- rollOff ל- 0.9. לאחר ההרצה תחזירו את ה- rollOff ל- 0.5.

13. תשוו את דיאגרמת העיין ואת ה- EVM שקיבלתם לתוצאה הקודמת. תסבירו מה אתם רואים ואיך קוראים לתופעה הזאת.

- תפעילו שוב את אפליקציית יצירת הסיגנל, תחזרו על כל השלבים של הגדרת האות (עם Rolloff factor = 0.5, אך הפעם תחת Impairments



תוסיפו רעש AWG של 15 dB (היות שה- SNR שהאפליקציה מוסיפה הוא עבור דגימה, לקבלת SNR לסימבול יש להשתמש בנוסחה $SNR(dB)_{sample} = SNR(dB)_{symbol} - 10\log(SpS)$.



14. תבדקו את הסיגנל בעזרת ה- VSA. מה ה- EVM שמתקבל? האם תואם את הציפיות?

- תיצרו את הסיגנל ותריצו את הקבצים של האלגוריתם.

15. תעיינו בדיאגרמת העיין ובדיאגרמת הקונסטלציה. האם מזהים את תוספת הרעש? מה ה- EVM המתקבל?

16. שאלת בנוס: תבנו קובץ אחד שפותר את בעיית סנכרון התדר ותיזמון הדגימות בעזרת עצמים מוכנים של MATLAB:

- comm.CoarseFrequencyCompensator
- comm.CarrierSynchronizer
- comm.SymbolSynchronizer

תשוו את התוצאה המתקבלת (עם אות סינטטי) לתוצאת האלגוריתם שהשתמשתם במהלך הניסוי.



נספחים

א' – תאור תוכן תיקיית Lab Files for students

התיקיה מכילה את כל הקבצים הנחוצים לביצוע הניסוי. חלק מהקבצים קוראים לפונקציות המצויות מתחת לקובץ הרלוונטי.

- Digital_Phase_Locked_Loop.m קובץ תרגול של חוג DPLL
 - Tx_Init.m קובץ איתחול של PLUTO
 - Capture_Rx_Pluto.m קובץ המעביר את האות הנקלט ב- PLUTO ל- MATLAB
 - Read_captured_Rx.m קובץ המתרגם את האות הנקלט ע"י PLUTO לפורמט MATLAB
 - Test_Signal.m קובץ שמיצר אות סינטטי לצורך בדיקת האלגוריתמים
 - CFC.m קובץ המבצע את האלגוריתם של Coarse Frequency Correction
 - Fine_carrier_sync.m קובץ המבצע את האלגוריתם של Fine Carrier Synchronizer
 - symbol_synchronizer.m קובץ המבצע את האלגוריתם של Symbol Synchronizer
 - MLTED_symTimingLoop.m פונקציה המממשת TED מסג Maximum Likelihood
 - ELTED_symTimingLoop.m פונקציה המממשת TED מסוג Early-Late
- כל אחת משתי הפונקציות הנ"ל נתמכת ע"י הפונקציה הבאה:
- timingLoopPIConstants.m קובץ המחשב את קבועי חוג ה- DPLL
 - מספר קבצים המסומנים כ- xx.p שהם קבצים מוסתרים מעיני הסטודנט (לרוב באורך של שורה או שתיים).

ב' – EVM

Error Vector Magnitude or **EVM** is a measure of how accurately a wireless system is transmitting symbols within its constellation. A digitally modulated wireless signal can be represented by a constellation diagram. The diagram displays the signal in the form of a scatter diagram in the xy plane. The angle of a point, measured counter-clockwise from the horizontal axis, represents the phase shift of the carrier wave from a reference phase. The distance of a point from the origin represents a measure of the amplitude (Volt) or power (Watt) of the signal.

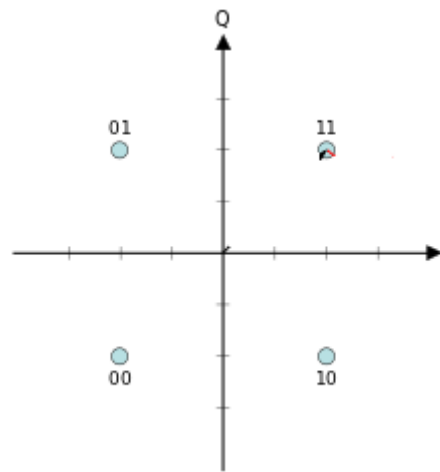


Figure 1-B QPSK constellation diagram

In an ideal world, a signal sent by a transmitter would have all the constellation points at the ideal locations, but in the real world, various imperfections cause the actual constellation points to deviate from their ideal location. These imperfections / errors include phase noise, low image rejection ratio, carrier leakage, etc.

EVM provides a comprehensive measure of the quality of the radio receiver or transmitter for use in digital communications by identifying sources of signal degradation, such as: filter distortion, amplitude non-linearity, I-Q imbalance, phase noise, etc. EVM is usually represented as a percentage but can also be measured in dB.

Let us consider a system (radio) that is transmitting via a QPSK constellation. Suppose, while transmitting the symbol [11], due to the presence of noise in the system, the received symbol deviates from its ideal position (the other symbols deviate as well). This deviation from the ideal constellation point and the point received by the receiver is the "**Error Vector**". The average amplitude of the error vector, normalized to the peak reference (ideal) signal amplitude is the "**Error Vector Magnitude**".

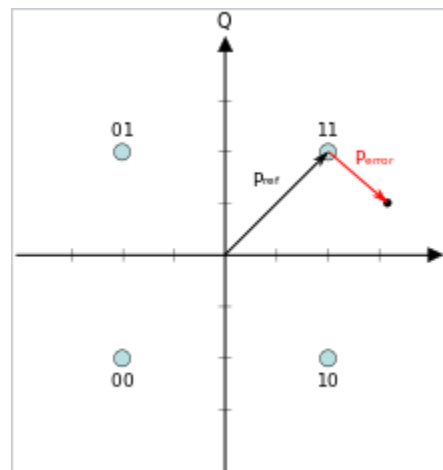


Figure 2-B Error of a symbol of QPSK Constellation

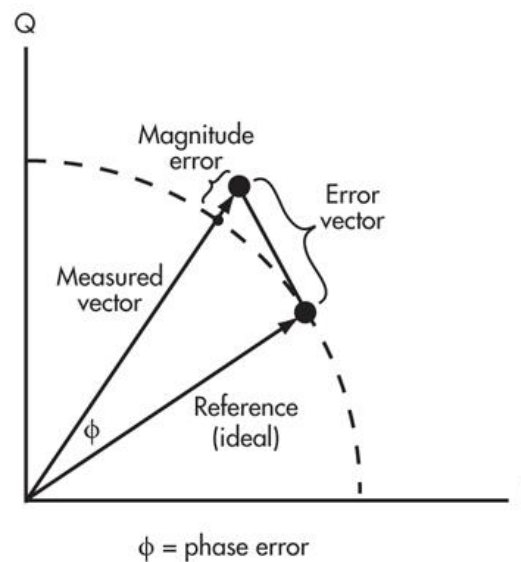


Figure 3-B IQ Plane Showing the Deviation of Measured Vector from the Ideal Vector



To calculate EVM in decibels (dB) we use the formula below (reference and error given as powers):

$$\text{EVM(dB)} = 10 \log_{10} \left(\frac{P_{\text{error}}}{P_{\text{reference}}} \right)$$

To calculate EVM as a percentage (%) we use the formula below:

$$\text{EVM}(\%) = \sqrt{\frac{P_{\text{error}}}{P_{\text{reference}}}} \times 100\%$$

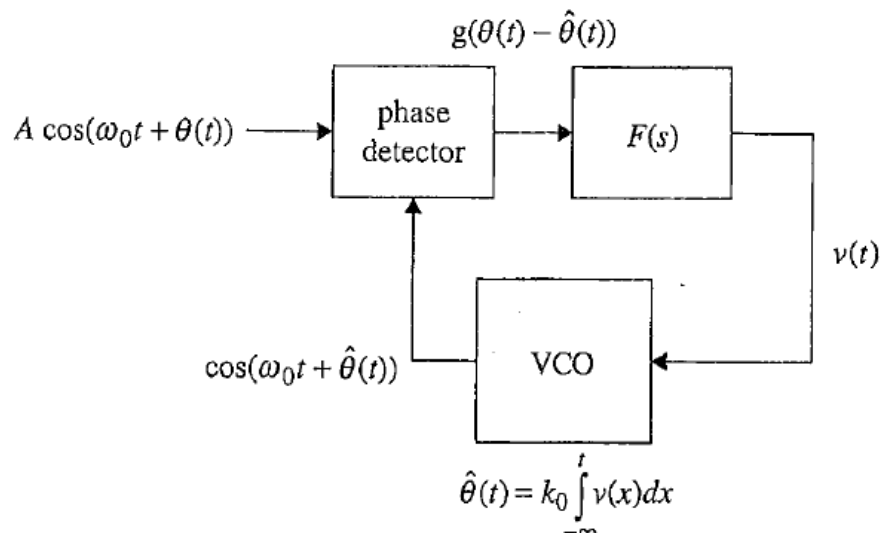
Can you figure out how to do the calculation if the reference and error are given as voltages?

ג' – DPLL

1. חוג PLL אנלוגי

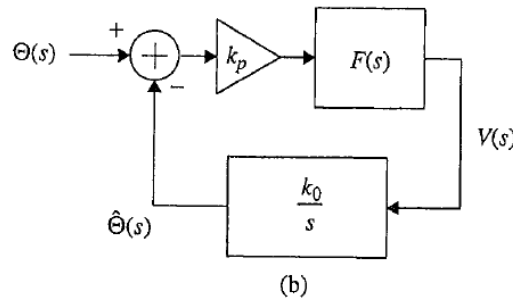
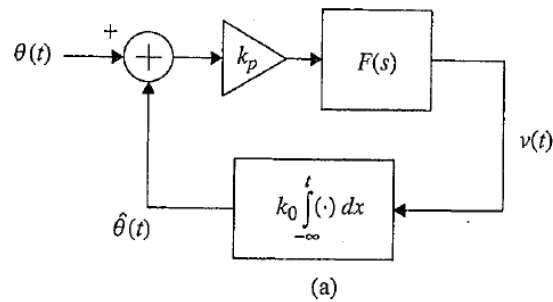
כדי להבין את פעולת ה-DPLL נעבור קודם על עיקרון פעולת PLL אנלוגי. תפקיד ה-PLL הוא לנעול (לסנכרן) את הפאזה (ותוך כדי כך התדר) של מתנד מבוקר מתח (VCO) לאות בניסה נתון. ה-PLL מורכב משלושה רכיבים:

- גלאי פאזה
- מסנן החוג
- מתנד מבוקר מתח



איור ג-1 מבנה בסיסי של PLL

ניתוח פעולת ה-PLL מבוצע בעזרת עקרונות תורת הבקרה. אפשר לנתח את החוג בתחום הזמן או בתחום התדר.



איור ג-2 תאור בתחום הזמן ותדר של חוג ה- PLL

המודלים של החוגים כוללים את הקבועים הבאים:

- k_p קבוע (הגבר) של גלאי הפאזה

- K_0 קבוע של ה- VCO

מסנן החוג המקובל לשימושים שלנו הוא מסוג proportional plus integrator:

$$F(s) = k_1 + \frac{k_2}{s}$$

כאשר משתמשים במסנן חוג כזה, החוג הופך לחוג בקרה מסדר שני שקיים עליו הרבה מידע.

התמרת Laplace של מוצא החוג נתונה ע"י:

$$\hat{\Theta}(s) = \frac{k_0 k_p F(s)}{s + k_0 k_p F(s)} \Theta(s)$$

במידה ומציבים את נוסחת מסנן החוג במשוואה הנ"ל, נקבל את פונקציית התמסורת של החוג:

$$H_a(s) = \frac{k_0 k_p k_1 s + k_0 k_p k_2}{s^2 + k_0 k_p k_1 s + k_0 k_p k_2}$$

את הנוסחה הזאת אפשר להשוות לפונקציית תמסורת מקובלת של חוג מסדר שני:

$$H_a(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

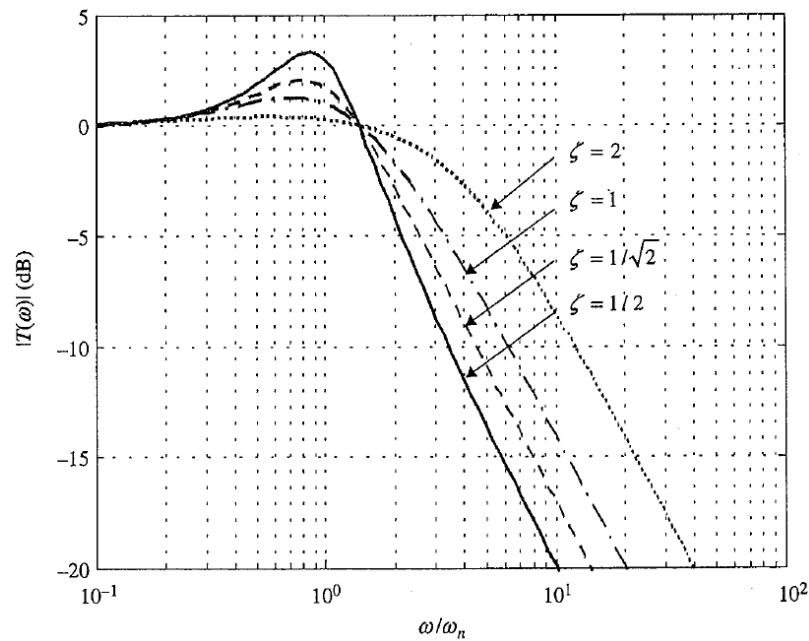


משתי הנוסחאות האלה אפשר לחלץ את מקדם הריסון ואת תדר האופייני של החוג:

$$\zeta = \frac{k_1}{2} \sqrt{\frac{k_0 k_p}{k_2}}$$

$$\omega_n = \sqrt{k_0 k_p k_2}.$$

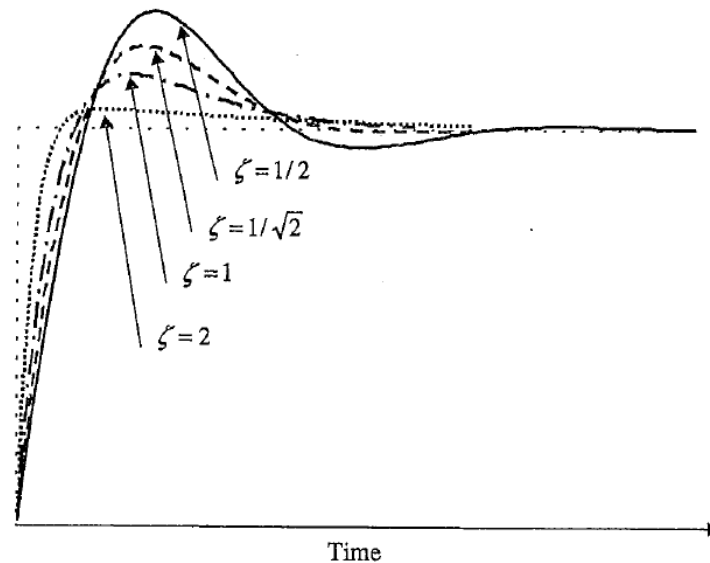
פונקציית התמסורת של ה-PLL כתלות במקדם הריסון נתון ע"י הגרף הבא:



איור ג-3 פונקציית תמסורת של חוג מסדר שני



התגובה לקפיצת מדרגה של פאזת בכניסת החוג נתונה בגרף הבא:



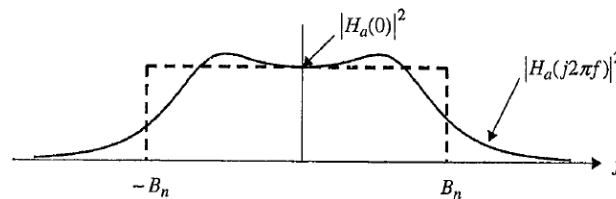
איור ג-4 תגובה לשינוי מדרגה של חוג מסדר שני

תכונות נוספות של חוג מסדר שני:

- רוחב סרט 3dB

$$\omega_{3dB} = \omega_n \sqrt{1 + 2\zeta^2 + \sqrt{(1 + 2\zeta^2)^2 + 1}}.$$

- רוחב סרט רעש (B_n) שווה ערך:



$$B_n = \frac{1}{2|H_a(0)|^2} \int_{-\infty}^{\infty} |H_a(j2\pi f)|^2 df.$$

$$B_n = \frac{\omega_n}{2} \left(\zeta + \frac{1}{4\zeta} \right)$$

מתוך כל משוואות הנ"ל אפשר לקבל את הביטויים הבאים שמהם אפשר לחשב את הפרמטרים של החוג:



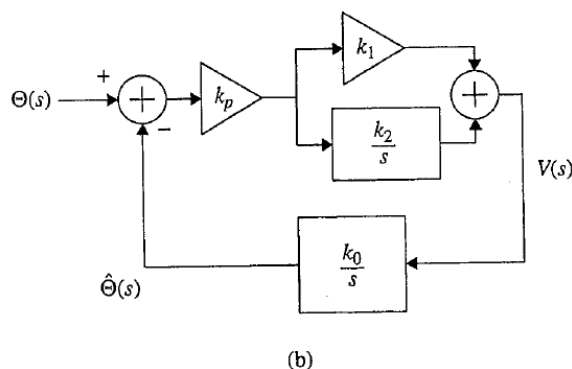
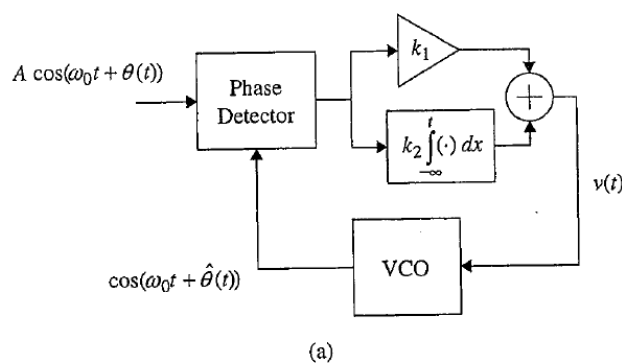
$$k_p k_0 k_1 = \frac{4\zeta B_n}{\zeta + \frac{1}{4\zeta}}$$

$$k_p k_0 k_2 = \frac{4B_n^2}{\left(\zeta + \frac{1}{4\zeta}\right)^2}$$

לרוב k_p ו- k_0 נתונים ולכן מהמשוואות הנ"ל אפשר לחשב את k_1 ו- k_2 .

2. חוג PLL דיגיטלי (DPLL)

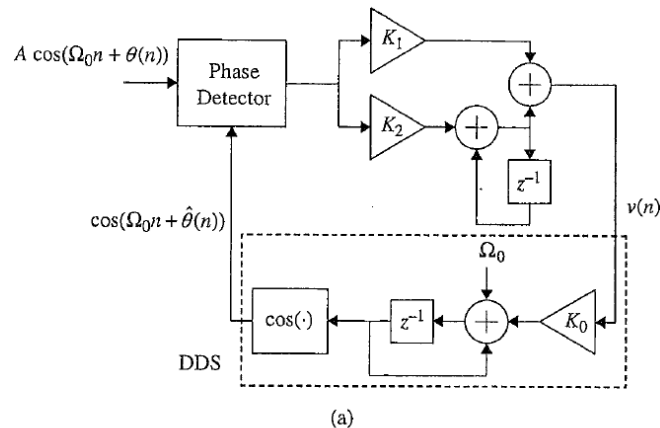
כפי שראינו עבור חוג PLL אנלוגי קיים מידע רב ולכן השיטה המקובלת לקבלת הנוסחאות עבור DPLL היא להמיר את הנוסחאות של PLL מהתמרת Laplace להתמרת Z. כזכור ניתן לתאר PLL אנלוגי מסדר שני בעל פילטר חוג מסוג proportional plus integrator בצורה הבאה:



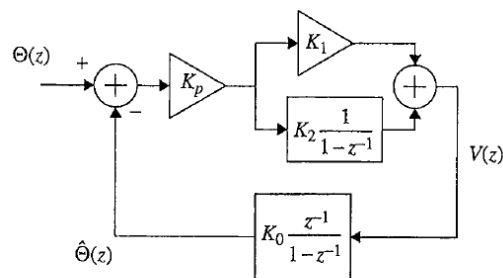
איור ג-5 חוג PLL בתחום הזמן ובתחום התדר



סכמות ה-DPLL המחקות את סכמות ה-PLL בזמן רציף הן הבאות:



(a)



(b)

איור ג-6 חוג DPLL המדמה חוג PLL בתיאור של זמן דיסקרטי ובתיאור לינארי של פאזה

רואים בבירור את הדמיון בין הסכמות, כאשר לכל חלק בסכמה האנלוגית יש את המקביל בסכמה הדיגיטלית. לדוגמה, המקביל שח ה-VCO הוא ה-DDS (Direct Digital Synthesizer).

פונקציות התמסורת עבור החוגים האנלוגי והדיגיטלי הן בהתאמה:

$$H_a(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$H_d(z) = \frac{K_p K_0 (K_1 + K_2) z^{-1} - K_p K_0 K_1 z^{-2}}{1 - 2\left(1 - \frac{1}{2} K_p K_0 (K_1 + K_2)\right) z^{-1} + (1 - K_p K_0 K_1) z^{-2}}$$

כרגע נפעיל את נוסחת ההמרה של Tustin בנוסחה של $H_a(s)$:

$$\frac{1}{s} \rightarrow \frac{T}{2} \frac{1 + z^{-1}}{1 - z^{-1}}$$



אחרי קצת עבודה אלגברית נקבל:

$$H_a \left(\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} \right) = \frac{\frac{2\zeta\theta_n + \theta_n^2}{1+2\zeta\theta_n + \theta_n^2} + 2 \frac{\theta_n^2 - \zeta\theta_n}{1+2\zeta\theta_n + \theta_n^2} z^{-1} + \frac{\theta_n^2}{1+2\zeta\theta_n + \theta_n^2} z^{-2}}{1 - 2 \frac{\theta_n^2 - 1}{1+2\zeta\theta_n + \theta_n^2} z^{-1} + \frac{1-2\zeta\theta_n + \theta_n^2}{1+2\zeta\theta_n + \theta_n^2} z^{-2}}$$

כאשר:

$$\theta_n = \frac{\omega_n T}{2}$$

בסופו של דבר, ע"י השווה האיברים של המשוואות בתחום z מקבלים:

$$K_p K_0 K_1 = \frac{4\zeta\theta_n}{1+2\zeta\theta_n + \theta_n^2}$$

$$K_p K_0 K_2 = \frac{4\theta_n^2}{1+2\zeta\theta_n + \theta_n^2}$$

כרגע אפשר לשלב את הנוסחאות:

$$\theta_n = \frac{\omega_n T}{2} \quad B_n = \frac{\omega_n}{2} \left(\zeta + \frac{1}{4\zeta} \right)$$

כדי לקבל:

$$\theta_n = \frac{B_n T}{\zeta + \frac{1}{4\zeta}}$$

הנוסחאות הסופיות לחישוב מקדמי החוג הן:

$$K_0 K_p K_1 = \frac{4\zeta \left(\frac{B_n T}{\zeta + \frac{1}{4\zeta}} \right)}{1 + 2\zeta \left(\frac{B_n T}{\zeta + \frac{1}{4\zeta}} \right) + \left(\frac{B_n T}{\zeta + \frac{1}{4\zeta}} \right)^2}$$

$$K_0 K_p K_2 = \frac{4 \left(\frac{B_n T}{\zeta + \frac{1}{4\zeta}} \right)^2}{1 + 2\zeta \left(\frac{B_n T}{\zeta + \frac{1}{4\zeta}} \right) + \left(\frac{B_n T}{\zeta + \frac{1}{4\zeta}} \right)^2}$$



במידה ורוחב סרט לרעש של החוג קטן בהרבה מקצב הדגימה (זהו המצב ברוב המקרים) אז בקירוב:

$$K_0 K_p K_1 \approx \frac{4\xi}{\xi + \frac{1}{4\xi}} (B_n T)$$

$$K_0 K_p K_2 \approx \frac{4}{\left(\xi + \frac{1}{4\xi}\right)^2} (B_n T)^2$$

לרוב K_0 ו- K_p ידועים ולכן ניתן לחשב את K_1 ו- K_2 .

מקורות

1. Gardner, Floyd M., Phaselock Techniques, 3rd Ed., Wiley-Interscience, 2005, Chapter 4.
2. Rice, Michael, Digital Communications, a Discrete-Time Approach, Pearson Prentice Hall, 2009, Appendix C.
3. R.W.Stewart, K.W. Barlee, D.S.W. Atkinson, and L.H. Crockett, Software Defined Radio using MATLAB & Simulink and the RTL-SDR, Published by Strathclyde Academic Media, 2015.
4. Fredric j harris, Multirate Signal Processing for Communication Systems, River Publishers; 2nd edition, 2021