להלן מדריך מקוצר להפעלת הקוד:

הדרך להפעיל ריצה אחת בודדת של זיהוי מטרות באמצעות המודל המסוים (סיגנל עם מודולציה בזמן) ושיטת הפתרון שלנו (מכפלת קרונקר):

שורת הריצה הינה:

[successVec,resultHist,realHist,targets,targets\_Coset] = sim1(Ci,Q,L,P,plot\_fail\_sim,numSims);

Inputs:

Ci => A vector with integers (Preferred to be prime). The number of integers will is T - the number of channels.

Q => An integer. Determine the ambiguity factor.

L => An integer. Determine the number of targets.

P => An integer. Determine the number of pulses in each cannel.

Plot\_fail\_sim => A Boolean. When true: plots a graph for a failed iteration. when false: doesn’t plot.

numSims => An integer. Determine the numbers of test iterations in the simulation.

Outputs:

successVec => A [numSims x2] matrix. Contain information about the <numSims> iterations in the simulation.

First column: The max distance of a real target and a result target (after detection)

Second column: number of the targets who were recovered successfully.

resultHist => An [numSims x L x 2] matrix. Contain for every iteration and every target the cell in the **recovered** space-velocity matrix.

realHist => An [numSims x L x 2] matrix. Contain for every iteration and every target the cell in the **original** space-velocity matrix.

Targets => A struct. Contains for every **original** target 3 numbers:

t – The target’s range (in time)

f – The target’s frequency (phase from the Doppler shift) – symbolizes the velocity

a – The signal amplitude – not in use.

targets\_Coset => A struct. Contains for every **recovered** target 3 numbers:

t – The target’s range (in time)

f – The target’s frequency (phase from the Doppler shift) – symbolize the velocity

a – The signal’s amplitude – not in use.

ברירת המחדל להרצת סימולציה:

[successVec,resultHist,realHist,targets,targets\_Coset] = sim1();

אשר שקולה להרצת הקוד הבא:

[successVec,resultHist,realHist,targets,targets\_Coset] = sim1([0 3 5 7 11 17 19 23 ],4,4,10,true,100);

בכל איטרציה בסימולציה אנו מבצעים את הסכמה הבאה:

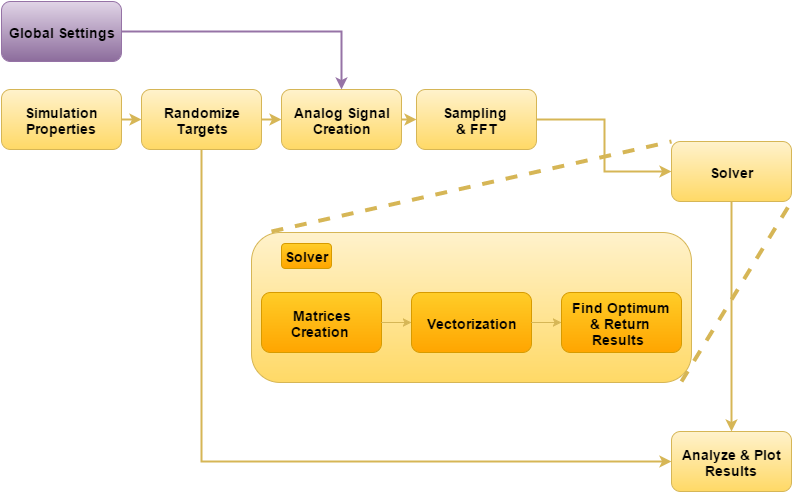


Figure 23

נציין ונרחיב מעט על הפונקציות העיקריות שאנו משתמשים בהן:

[g] = global\_settings(P, P, L, Ci,Q)

The inputs are transferred from the sim1 input or the default values specified above.

The output is a struct that contain all kind of constants for the simulation.

[targets] = randomize\_targets(g)

Taking the global settings and return L randomize targets with a certain distance and velocity.

The struct targets are like specified above.

[x] = generate\_analog\_input\_signal(g\_coset, targets)

Taking the randomize target and the global settings and create (the output) an analog RX signals (here we can see our modulation).

[targets\_kron] = coset\_nyquist(g, x, targets)

Inputs:

G – global settings

X – the analog RX signal – a matrix, each row is a bucket.

Targets – the original randomize target – for equation check usage.

Output:

The recovered targets. Specified above.

[success,realHist,resultHist,successVec] = analyze\_result (g\_coset,targets,targets\_Coset,iterNum,plot\_fail\_sim)

Inputs:

g\_coset - global settings

targets – original targets

targets\_Coset – recovered targets

iterNum – iteration number (the simulation is from several runs for statistic)

flot\_fail\_sim – like in sim1();

Outputs:

Success – 0 if run failed, 1 if successful.

realHist,resultHist,successVec – like in sim1().

לשאלות נוספות על הפעלת הקוד, ניתן לפנות לאחד מכתובות הדוא"ל הבאות:

[shanlior@gmail.com](mailto:shanlior@gmail.com)

gal\_winerich@hotmail.com