**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI.NO** |  | **TITLE** | **PAGE NO.** |
|  |  | ABSTRACT | 2 |
| 1 |  | INTRODUCTION | 4 |
| 2 |  | VEHICLE MAPPING SYSTEM | 5-8 |
|  | 2.1  2.2  2.3 | SOFTWARE SPECIFICATIONS  BLOCK DIAGRAM OF SYSTEM  METHADOLOGY USED | 5  6  7 |
| 3 | 3.1  3.2  3.3 | IMPLEMENTATION OF THE PROJECT  MATLAB CODE  EXPLANATION OF THE CODE  MERITS AND DEMERITS | 9-14  9  12  14 |
| 4 |  | CONCLUSION OF THE PROJECT | 15 |
|  | 4.1 | CONCLUSION | 15 |
|  | 4.2 | FUTURE WORK OF THE PROJECT | 15 |

**CHAPTER-1**

**INTRODUCTION**

After the advent of technologies the automation and usage of internet and electricity has drastically increased. Even many disasters happening everyday and it is inevitable. One of the disasters is accidents, happening while transportation to reduce it we are using FMCW technology based mapping system. In chapter 2, the software specification, block diagram and methodology used is explained. In chapter 3, the code, its explanation and merits, demerits is explained. In chapter 4, the future work of the project is explained.

**CHAPTER 2**

**VEHICLE MAPPING SYSTEM**

This chapter includes the methodology, software specifications, explanation of the block diagram and the working of the system.

**2.1 SOFTWARE SPECIFICATIONS**

The software used to create this system is MATLAB which is a high-performance language for technical computing.

It is a proprietary multi-paradigm programming language and numeric computing environment developed by Math Works. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithm, creation of user interfaces, and interfacing with programs written in other languages [1].

Although MATLAB is intended primarily for numeric computing, an optional toolbox used the MuPAD symbolic engine allowing access to symbolic computing abilities. An additional package, simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

MATLAB supports structure data types. Since all variables in MATLAB are arrays, a more adequate name is “structure array”, where each element of the array has the same field names. In addition, MATLAB supports dynamic field names (field look-ups by name, field manipulations, etc.).

When creating a MATLAB function, the name of the file should match the name of the first function in the file. Valid function names begin with an alphabetic character, and can contain letters, numbers, or underscores. Variables and functions are case sensitive.

It supports object-oriented programming including classes, inheritance, virtual dispatch, packages, pass-by-value semantics, and pass-by-reference semantics. However, the syntax and calling conventions are significantly different from other languages. MATLAB has value classes and reference classes, depending whether the class has handle as a super-class for reference classes or nor for value classes.

**2.2 BLOCK DIAGRAM OF THE SYSTEM**

1. At first the signal is transmitted and received after hitting a target.
2. If a target is not identified it won’t under go any other operation.
3. If a target is detected, it will go for visualizing process.

WORKING OF RADAR:

Extremely short bursts of radio energy (traveling at the speed of light) are transmitted, reflected off a target and then returned as an echo. Radar makes use of a phenomenon we have all observed, that of the echo principle [2].

Same like RADAR instead of normal waves it transmits a high frequency signal whose frequency increases linearly during the measurement phase (called the frequency sweep). The signal is emitted, reflected from the measuring surface and received with a time delay, t [3].

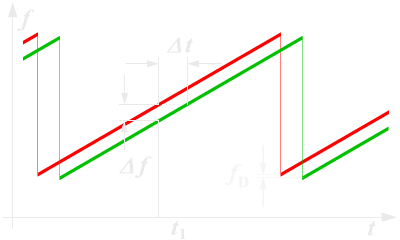
**2.3 METHADOLOGY USED**

DEFINITION OF FMCW:

FMCW radar (Frequency-Modulated Continuous Wave radar = FMCW radar) is a special type of radar sensor which radiates continuous transmission power like a simple continuous wave radar ([CW-Radar](https://www.radartutorial.eu/02.basics/Continuous%20Wave%20Radar.en.html)). In contrast to this CW radar FMCW radar can change its operating frequency during the measurement: that is, the transmission signal is modulated in frequency (or in phase). Possibilities of Radar measurements ​​through runtime measurements are only technically possible with these changes in the frequency (or phase) [4].

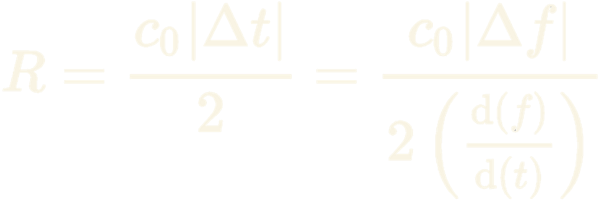
WORKING OF FMCW:

A signal is transmitted, which increases or decreases in the frequency periodically. When an echo signal is received, that change of frequency gets a delay Δt (by runtime shift) like to as the pulse radar technique. In pulse radar, however, the runtime must be measured directly. In FMCW radar are measured the differences in phase or frequency between the actually transmitted and the received signal instead.



MEASURING DISTANCE:

1. The distance measurement is accomplished by comparing the frequency of the received signal to a reference (usually directly the transmission signal).
2. The duration of the transmitted waveform *T* is substantially greater than the required receiving time for the installed distance measuring range.



*c0* = speed of light = 3·108 m/s

*Δt* = delay time [s]

*Δf* = measured frequency difference [Hz]

*R* = distance between antenna and the reflecting object (ground) [m]

*df/dt* = frequency shift per unit of time

FEATURES OF FMCW:

* Ability to measure very small ranges to the target (the [minimal measured range](https://www.radartutorial.eu/01.basics/Minimal%20Measuring%20Range.en.html) is comparable to the transmitted wavelength).
* Ability to measure simultaneously the target range and its relative velocity.
* Very high accuracy of range measurement.
* Signal processing after mixing is performed at a low frequency range, considerably simplifying the realization of the processing circuits.
* Safety from the absence of the pulse radiation with a high peak power.

**CHAPTER-3**

**IMPLEMENTATION OF THE PROJECT**

**3.1 MATLAB CODE**

clc

fc = 77e9;

c = 3e8;

range\_max = 200;

% sweep time should be 5-6 times the rtt

time = 5.5\*range2time(range\_max,c); %Range between is converted from range to time

%range resolution which will help us to distinguish between target

range\_res = 1;

%the range2bw gies the band wisth bearing the rannge resolution ,c refers the propagation speed

bw = range2bw(range\_res,c);

sweep\_slope = bw/time;

%linear FMCW -> Beat frequency

fr\_max = range2beat(range\_max,sweep\_slope,c);

% Maximum speed of vechicle = 150km/hr -> m/s

lambda = c/fc;

v\_max = 150\*10/36;

fd\_max = speed2dop(2\*v\_max,lambda); % Dopple frequency

fb\_max = fr\_max+fd\_max; % Maximum beat frequnecy

fs = max(2\*fb\_max,bw); %150MHz

waveform = phased.FMCWWaveform('SweepTime',time,'SweepBandwidth',bw,'SampleRate',fs);

sig = waveform();

subplot(211); plot(0:1/fs:time-1/fs,real(sig));

xlabel('Time (s)');

ylabel('Amplitude (v)');

title('FMCW signal');

subplot(212);

spectrogram(sig,32,16,32,fs,'yaxis');

%Dividing the signal into sections of length 32.

%16 refers the samples used.

%Evaluating the spectrum at 32/2+1=17 frequencies

title('FMCW signal spectrum');

% Simulation

car\_dist = 70;

car\_speed = 96;

car\_rcs = db2pow(min(10\*log10(car\_dist)+5,20)); %Radar Cross Section where the db is converted to power

%For fixing a target and to choose whether polaraized or non polaraized

%Instead of mean RCS we can use polaraization as TRUE but we use radar

%cross section we simulated to get diserable output

cartarget = phased.RadarTarget('MeanRCS',car\_rcs,'PropagationSpeed',c,'OperatingFrequency',fc);

%For modeling the translational motion of objects in space

carmotion = phased.Platform('InitialPosition',[car\_dist;0;0.5],'Velocity',[car\_speed;0;0]);

%For narrow band signal propagation from one point to another

channel = phased.FreeSpace('PropagationSpeed',c,'OperatingFrequency',fc,'SampleRate',fs,'TwoWayPropagation',true);

%SYSTEM SETUP

ant\_aperture = 6.06e-4; % in square meter

ant\_gain = aperture2gain(ant\_aperture,lambda); % in dB

tx\_ppower = db2pow(5)\*1e-3; % in watts

tx\_gain = 9+ant\_gain; % in dB

rx\_gain = 15+ant\_gain; % in dB

rx\_nf = 4.5;

% The receiver will have some noise factor that we are mentioning so it could more like a real world simulation

transmitter = phased.Transmitter('PeakPower',tx\_ppower,'Gain',tx\_gain);

receiver = phased.ReceiverPreamp('Gain',rx\_gain,'NoiseFigure',rx\_nf,'SampleRate',fs);

% SIGNAL SETUP

radar\_speed = 100;

radarmotion = phased.Platform('InitialPosition',[0;0;0.5],'Velocity',[radar\_speed;0;0]);

specanalyzer = dsp.SpectrumAnalyzer('SampleRate',fs,'PlotAsTwoSidedSpectrum',true,'Title','Spectrum for received and dechirped signal','ShowLegend',true);

rng(); %random number generator

% The Received signal is dechirped and saved in a buffer and 64 is the number of sweeps

% we can even use 8,16,32 sweeps but for better accuracy 64 sweeps is used

Nsweep = 64;

demo = complex(zeros(waveform.SampleRate\*waveform.SweepTime,Nsweep));

for n = 1:Nsweep

% Update radar and target positions

[radar\_pos,radar\_vel] = radarmotion(waveform.SweepTime);

[target\_pos,target\_vel] = carmotion(waveform.SweepTime);

% Transmit FMCW waveform

sig = waveform();

txsig = transmitter(sig);

% Propagate the signal and reflect off the target

txsig = channel(txsig,radar\_pos,target\_pos,radar\_vel,target\_vel);

txsig = cartarget(txsig);

% Dechirp the received radar return

txsig = receiver(txsig);

dechirpsig = dechirp(txsig,sig);

% Visualize the spectrum

specanalyzer([txsig dechirpsig]);

%channel-1 = received signal

%channel-2 = decyphered signal

demo(:,n) = dechirpsig;

end

rngdopresp = phased.RangeDopplerResponse('PropagationSpeed',c,'DopplerOutput','Speed','OperatingFrequency',fc,'SampleRate',fs,'RangeMethod','FFT','SweepSlope',sweep\_slope,'RangeFFTLengthSource','Property','RangeFFTLength',2048,'DopplerFFTLengthSource','Property','DopplerFFTLength',256);

plotResponse(rngdopresp,demo); % Plot range Doppler map

axis([-v\_max,v\_max,0,range\_max])

clim = caxis;

%Reducing the sampling rate and estimating the distance and velocity

Dn = fix(fs/(2\*fb\_max));

for m = size(demo,2):-1:1

demo1(:,m) = decimate(demo(:,m),Dn,'FIR');

end

fs1 = fs/Dn;

fbrange = rootmusic(pulsint(demo1,'coherent'),1,fs1); %finding range with beat frequency

rangeest = beat2range(fbrange,sweep\_slope,c); %Range estimation

peak\_loc = val2ind(rangeest,c/(fs1\*2));

fd = -rootmusic(demo1(peak\_loc,:),1,1/time); %Estimates the frequency in input signal and gives vector of frequerncy in radian / samples

velocityest = dop2speed(fd,lambda)/2;

**3.2 EXPLANATION OF THE CODE:**

Range2time:

In range2time(r,c) r is range mentioned which is in meters and c is propagation speed which is considered as speed of light [5]. In order measure the distance between two objects it returns the time a signal takes to propagate a given distance.

Range2bw:

In range2bw(res,c) res mentions the range resolution and as before c signifies the speed of light. It returns the bandwidth needed to distinguish two targets separated by a given range [6].

Range2beat:

In range2beat(r,sweepslope,c) it converts the range of dechirped linear FMCW signal to the corresponding beat frequency, sweep slope is mentioned in the FMCW figure [7].

Speed2dop:

In this function we need to mention the radial velocity and wavelength in order to get the Doppler shift in hertz.

Waveform:

For obtaining waveform phased.FMCW wavefrom function is used. In this function Sweep time, Sweep bandwidth and sampling frequency is mentioned [8].

Spectrogram:

It computes the short-time fourier transform of a signal. The spectrogram is the magnitude of this function [9]. In this function we need to mention the length of the sections of the signal, the samples used and evaluating spectrum frequencies.

Target:

The quantity that determines the response of a target to incoming signals is called the radar target cross-section (RCS). While all electromagnetic radar signals are polarized, you can sometimes ignore polarization and process them as if they were scalar signals. To ignore polarization, specify the enable polarization function as false [10].

Platform:

This function models the translational motion of one or more platforms in space. A platform can be a target such as a vehicle, or radar transmitter and receiver. The model assumes that the platform undergoes translational motion at constant velocity or constant acceleration during each simulation.

Aperture2gain:

The function returns the antenna gain in decibels corresponding to an effective aperture of square meters for an incident electromagnetic wave with wavelength lambda. The given aperture can be scalar or vector [11].

Doppler response:

Phased.RangeDoppleResponse, It calculates the filtered response to fast-time and slow-time or equivalent, range data using either matched filter or an FFT.

Root music:

It estimates the frequency content in the input signal and returns a vector of frequencies in radian/sample. The extra threshold parameter win the second entry in p provides you more flexibility and control in assigning the noise and signal subspaces.

**3.3 MERITS AND DEMERITS**

MERITS:

1. It is cost efficient when comparing with RADAR.
2. It uses for low power transmission.
3. By using FMCW higher bandwidth can be obtained,
4. When comparing with CW, FMCW can frequently change frequency on accord to it.

DEMERITS:

1. It can be used for target at short range because of lower peak output power.
2. Due to low transmit power, the signal gets attenuated and affected due to atmosphere and channel before it is received by the receiver.
3. The transmission has a probability of getting interfered by other nearby radio systems due to use of large range of frequencies and low peak power for transmission.
4. The maximum range is limited by the power that it can radiate.

**CHAPTER-4**

**CONCLUSION OF THE PROJECT**

**4.1 CONCLUSION**

In this project we created a cost efficient mapping system which is able to track an obstacle distance and speed if it is a moving system based on FMCW, which can be used any mode of transport system except airways.

**4.2 FUTURE WORK OF THE PROJECT**

1. As for now it can analyze a single obstacle in front of it, but we are planning to make it track multiple objects.

2. Even though the range is satisfying it cannot be increased based on FMCW. So, even efficient method can be handled for increasing range.

3. The transmission power it used is less but even though it is not possible for always and by using higher transmission power the range can also be extended.

4. This project is done in software level and for implementing in real life, we will be trying to use sensors and microcontrollers.

5. if we increase the ranges it will be possible for implementing this even in air transport,

**REFERENCES:**

[1] <https://en.wikipedia.org/wiki/MATLAB>

[2]<https://msi.nga.mil/api/publications/download?key=16694476/SFH00000/310ch1.pdf>

[3] <https://instrumentationtools.com/fmcw-radar-level-transmitter-working-principle/>

[4]<https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html>

[5]<http://www.gusucode.com/index.php?m=content&c=index&a=show&catid=34&id=12103&file_name=phased/range2time.m>

[6] <https://in.mathworks.com/help/phased/ref/range2bw.html>

[7] <https://in.mathworks.com/help/phased/ref/range2beat.html>

[8] <https://in.mathworks.com/help/phased/ref/phased.fmcwwaveform-system-object.html>.

[9] <http://matlab.izmiran.ru/help/toolbox/signal/spectrogram.html>

[10] <https://in.mathworks.com/help/phased/ref/phased.radartarget-system-object.html>

[11] <https://www.mathworks.in/help/phased/ref/aperture2gain.html#:~:text=(A%2Clambda)-,Description,be%20a%20scalar%20or%20vector>.