

جامعة دمشق

كلية الهندسة الميكانيكية والكهربائية

قسم هندسة الحواسيب والأتمتة



# تطوير خوارزميات تعلّم الآلة في نظم تحديد وتتبع الموقع

مشروع تخرج أعد لنيل درجة الإجازة في هندسة الحواسيب والأتمتة

إعداد الطالب

**عبدالله السعيد**

بإشراف

**د.م. هيام خدام**

العام الدراسي

2018-2017

## الإهداء

إلى عائلتي أُمي... أبي... أخوتي

من لم يبخل علي بالدعم خالتي وخالي

أصدقائي وأخوتي وأخص منهم

أنس.. محمد ملاطيه لي... رضوان... ياسر... بشر... علي.. لؤي

إلى كل طالب علم هذا المشروع مفتوح المصدر والشيفرة المصدرية له موجودة

على موقع Github في حسابي [abdullahalsaidi16](#)

# كَلِمَةُ شُكْرِ

كلمة شكر أقل ما يقال لدكتورة المهندسة هيام خدام التي أشرفت علمياً على جميع

مراحل هذا المشروع

إلى الصديق المهندس يحيى صالح لدعمه المستمر

إلى من حمل رسالة العلم بأمانة أخص منهم:

الدكتور رؤوف حمدان

المهندس حسن نوفل

المهندسة مايا تقي

المهندس ماهر نديوي

إلى شركة سيريتل لتعاونها واهتمامها

إلى من نشر العلم بإنسانية Prof. Andrew Ng

إلى من ساعدنا بعلمه وعمله في هذا الطريق أخص منهم

لؤي صهيون... خالد دلة ... محمد بكر جليلاتي .. محمد باكير ... رنا حلي ... ريما ظواهرة ... فريق

Alpha... فريق RBT

**والحمد لله رب العالمين**

## ملخص المشروع

يهدف المشروع إلى تقديم توجه جديد للاستفادة من تطور قدرات خوارزميات تعلم الآلة في نظم تحديد الموقع، يعتمد على استثمار مجموعة الإشارات اللاسلكية من خلال تقديم دراسة بحثية تشمل الربط بين شدّات الإشارة المستقبلية والموقع الموافق ومن ثم الحصول على تنبؤ أمثلي عن طريق مرشح كالمان، بافتراض أن حركة الهدف وعملية القياس هما عمليتان عشوائيتان Stochastic Processes.

ويتضمن المشروع تطبيق عملي يشمل تطبيق النظام ببيئة خارجية Outdoor ، تشمل منطقة ضمن مدينة دمشق، بالتعاون مع شركة سيريتل، بالإضافة إلى تطبيق على بيئة داخلية ضمن بناء هندسة الحواسيب والأتمتة في كلية الهندسة الميكانيكية والكهربائية بواسطة Mobile Robot.

## فهرس الأشكال

- الشكل 1—1 بنية شبكة الحساسات اللاسلكية ..... 19
- الشكل 1—2 تحديد الموضع حسب على حساس القرب ..... 20
- الشكل 1—3 تحديد الموضع حسب AoA ..... 21
- الشكل 1—4 المبدأ الأساسي لطريقة تحديد الموقع عن طريق شدة الإشارة المستقبلية ..... 22
- الشكل 1—5 توضع المصنف ذو المسافة الأكبر ..... 24
- الشكل 1—6 فكرة المستوي الفائق الأمثلي ..... 26
- الشكل 1—7 نقاط أشعة الدعم Support Vectors ..... 26
- الشكل 1—8 نقاط الضجيج وانتهاكها لمنطقة الفصل ..... 32
- الشكل 1—9 تصنيف النماذج غير قابلة خطيا للفصل عن طريق Kernel ..... 34
- الشكل 1—10 مبدأ عمل Random Forest ..... 37
- الشكل 1—11 مراحل خوارزمية K-Means ..... 40
- الشكل 1—12 طريقة elbow ..... 41
- الشكل 1—13 الانحياز والتباين والانحياز والنمذجة المناسبة ..... 42
- الشكل 2—1 الطبيعة العودية لمرشح كالمان ..... 50
- الشكل 2—2 مقدار الخطأ في تقدير الموقع لكل من عملية القياس (باللون الأزرق)، وعملية القياس بعد استخدام مرشح كالمان (باللون الأخضر) ..... 56
- الشكل 2—3 موضع السيارة الحقيقي، المقاس، والنتائج عن مرشح كالمان. يظهر اللون الأخضر في الصورة التالية القيم المقاسة لموضع السيارة، فيما يظهر اللونان الأزرق والأحمر القيم الحقيقية والمتوقعة ..... 57
- الشكل 2—4 مقدار الخطأ في قياس السرعة ..... 57
- الشكل 2—5 القيم الحقيقية للسرعة والقيم المتنبئ بها عن طريق مرشح كالمان ..... 57
- الشكل 3—1 أبعاد البيئة وأماكن تموضع الحساسات ..... 62
- الشكل 3—2 توزع نقاط التدريب ضمن البيئة وأماكن الحساسات ..... 63
- الشكل 3—3 المسار الحقيقي المفروض ضمن البيئة ..... 64
- الشكل 3—4 توقعات المسار بحسب خوارزمية تحديد الموقع ..... 65
- الشكل 3—5 ترشيح خرج خوارزمية تحديد الموقع ..... 65
- الشكل 3—6 المخطط الصندوقي لتحديد الموقع عن طريق SVM ..... 66
- الشكل 3—7 المخطط الصندوقي لتحديد الموقع عن طريق RFF ..... 66

الشكل 3—8	المسار الأول الحقيقي والمتوقع من SVM و RFF بعد استخدام مرشح كالمان من الدرجة الأولى.....	68
الشكل 3—9	المسار الأول الحقيقي والمتوقع من SVM و RFF بعد استخدام مرشح كالمان من الدرجة الثانية.....	68
الشكل 3—10	المسار الثاني الحقيقي والمتوقع من SVM و RFF بعد استخدام مرشح كالمان من الدرجة الأولى.....	69
الشكل 3—11	المسار الثاني الحقيقي والمتوقع من SVM و RFF بعد استخدام مرشح كالمان من الدرجة الثانية.....	69
الشكل 3—12	المسار الثالث الحقيقي والمتوقع من SVM و RFF بعد استخدام مرشح كالمان من الدرجة الأولى.....	70
الشكل 3—13	المسار الثالث الحقيقي والمتوقع من SVM و RFF بعد استخدام مرشح كالمان من الدرجة الثانية.....	70
الشكل 3—14	الخطا الوسطي المطلق بدلالة عدد عناصر RFF.....	71
الشكل 3—15	منحنيات التعلم للخطأ الوسطي المطلق لخوارزمية RFF و Kalman Filter.....	71
الشكل 3—16	مبدأ عمل نظام تحديد الموقع.....	72
الشكل 3—17	أماكن استحصال العينات.....	73
الشكل 3—18	بعض مصادر استهلاك الطاقة في أجهزة Android من مؤتمر Google I/O 2017.....	74
الشكل 3—19	المخطط التدفقي لأنبوب تعلم الآلة.....	75
الشكل 4—1	عمل خوارزمية SLAM.....	79
الشكل 4—2	المخطط الصندوقي ل SLAM.....	81
الشكل 4—3	المبدأ النظري لإشارات حساس المسافة مع نقاط علام.....	82
الشكل 4—4	الشكل الروبوت ونقاط العلام.....	83
الشكل 4—5	إشارات حساس المسافة.....	83
الشكل 4—6	توقع بحسب Odom.....	84
الشكل 4—7	فرق بين الموقع المتوقع والحقيقي.....	84
الشكل 4—8	مراحل مرشح كالمان في SLAM.....	86
الشكل 5—1	شعار Android.....	90
الشكل 5—2	المكونات الأساسية لنظام تشغيل Android.....	92
الشكل 5—3	شعار Android Studio.....	93
الشكل 5—4	متطلبات Android Studio.....	94
الشكل 5—5	دورة حياة الفعالية.....	97
الشكل 5—6	دورة حياة الخدمة.....	99
الشكل 6—1	شعار ROS.....	102
الشكل 6—2	آلية إرسال الرسائل بين العقد.....	106
الشكل 6—3	آلية التواصل بين مخدم وعميل.....	107

108.....	الشكل 6—4 Action بين مخدم وعميل في ROS
109.....	الشكل 6—5 إصدارات ROS
110.....	الشكل 6—6 إعدادات تنصيب على Flash memory
112.....	الشكل 6—7 آلية تواصل بين العقد في ROS
113.....	الشكل 6—8 حساسات ممكن استخدامها في ROS
114.....	الشكل 6—9 أنواع Arduino
116.....	الشكل 6—10 STM32
116.....	الشكل 6—11 TI Launchpads
117.....	الشكل 6—12 لوحة Raspberry Pi
117.....	الشكل 6—13 أنواع ODRIOD
118.....	الشكل 6—14 شعار GAZEBO
119.....	الشكل 6—15 احدى بيئات Gazebo
120.....	الشكل 6—16 شعار برنامج RVIZ
123.....	الشكل 7—1 أنواع Turtlebot3
124.....	الشكل 7—2 مكونات Turtlebot3 burger
125.....	الشكل 7—3 حساس الليزر 360
127.....	الشكل 7—4 Raspberry Pi
130.....	الشكل 7—5 الوحدات المحيطية Raspberry Pi
131.....	الشكل 7—6 واجهة برنامج VNC Client ومكان كتابة IP الخاص بـ Raspberry Pi
132.....	الشكل 7—7 لوحة OpenCR
135.....	الشكل 7—8 محركات Dynamixel
139.....	الشكل 8—1 مخطط عمل نظام تحديد الموقع
140.....	الشكل 8—2 المعلومات الأساسية لبناء النظام
141.....	الشكل 8—3 مخطط rqt graph
142.....	الشكل 8—4 واجهة إدخال Master IP
143.....	الشكل 8—5 جزء من ملف Manifest
143.....	الشكل 8—6 ملف csv المنشأ بعد إغلاق عقدة data_formatter
145.....	الشكل 8—7 تقسيم الخريطة إلى عنايد
146.....	الشكل 8—8 يبين خرج عقدة تحديد الموقع

## فهرس الجداول

الجدول 3—1	الخطأ الوسطي المطلق لخوارزمية SVM	67
الجدول 3—2	الخطأ الوسطي المطلق لخوارزمية RFF	67
الجدول 6—1	مواصفات لوحة Arduino	115
الجدول 7—1	مواصفات Turtlebot3	125
الجدول 7—2	مواصفات Raspberry Pi 3 B	127
الجدول 7—3	مواصفات OpenCR	133
الجدول 7—4	مواصفات محرك Dynamixel XL-430 W 250	135



## جدول المصطلحات

أ	
Feature Extraction	استخلاص السمات
Support Vector Machine(SVM)	آلة أشعة الدعم
ت	
Simultaneous Localization and Mapping (Slam)	تحديد الموقع وبناء الخريطة بشكل متزامن
Feature Mapping	تحويل السمات
Fourier Transform	تحويل فورييه
Unsupervised Learning	التعلم غير المشرف عليه
Supervised Learning	التعلم المشرف عليه
Kernel Approximation	تقريب النواة
ج	
Nearest Neighbors	الجيران الأقرب
ح	
Kernel Trick	حيلة النواة
ذ	
Artificial Intelligence	الذكاء الصناعي
س	
Random Fourier Features (RFF)	سمات فورييه العشوائية
ش	
Decision Tree	شجرة قرار
Support Vector (SV)	شعاع الدعم
ض	
Process Noise	ضجيج العملية
Measurement Noise	ضجيج القياس
ع	
Shift Invariant	عدم التأثير بالانزياح العددي
Stochastic Process	العمليات العشوائية
Cluster	عنقود
غ	
Random Forest	غابة عشوائية

ف	
Feature Space	فضاء السمات
ل	
Structured Query Language (SQL)	لغة الاستعلام الهيكلية
م	
Kalman Filter	مرشح كالمان
Particle Filter	مرشح الجسيمات
Extended Kalman Filter	مرشح كالمان الممدد (الخطي)
Soft Margin	مسافة الفصل التي تقبل وجود نقاط ضجيج
Margin	هامش الفصل بين الأصناف
Received Signal Level (RSL)	مستوى الإشارة المستقبلية
Covariance Matrix	مصفوفة التباين
Odometry	مقدار التغير في الموقع مع الزمن
Received Signal Strength Indicator (RSSI)	مؤشر قوة الإشارة المستقبلية
ن	
Global System for Mobile Communication (GSM)	النظام العالمي للاتصالات الهواتف النقالة

## فهرس المحتويات

5	فهرس الأشكال
8	فهرس الجداول
9	جدول المصطلحات
16	مقدمة
17	الفصل الأول
17	1 شبكة الحساسات اللاسلكية وخوارزميات تعلم الآلة
18	1.1 مقدمة
18	1.2 شبكة الحساسات اللاسلكية
18	1.2.1 بنية الشبكة اللاسلكية
19	1.2.2 تحديد الموقع ضمن شبكة الحساسات اللاسلكية
23	1.3 خوارزميات تعلم الآلة
23	1.3.1 خوارزمية Support Vector Machine
35	1.3.2 خوارزمية سمات فورييه العشوائية Random Fourier Features
37	1.3.3 خوارزمية Random Forest
39	1.3.4 خوارزمية K-Means
42	1.3.5 التوازن بين الانحياز والتباين
43	الفصل الثاني
43	2 العمليات العشوائية ومرشح كالمان
44	2.1 مقدمة
45	2.2 مفهوم العمليات العشوائية
46	2.3 نموذج النظام الحتمي

47	2.4 نموذج النظام العشوائي.....
47	2.5 مرشح كالمان الخطي.....
47	2.5.1 مفهوم مرشح كالمان وآلية عمله.....
51	2.6 مرشح كالمان غير الخطي (EKF) Extended Kalman Filter.....
54	2.7 مثال عددي.....
58	الفصل الثالث.....
	3الدراسة الأمثلية لتحديد الموقع وتطبيق عملي على معطيات شركة الاتصالات
58	سيريتل(Syriatel).....
59	3.1 مقدمة.....
59	3.2 التعريف بأهم المكتبات المستخدمة.....
60	3.3 دراسة الأمثلية لخوارزميات تعلم الآلة ضمن شبكة الحساسات اللاسلكية.....
60	3.3.1 منهجية العمل.....
62	3.3.2 نموذج أوكومورا-هاتا Okumura-Hata Model.....
63	3.3.3 إنشاء المسارات الافتراضية.....
64	3.3.4 تحديد الموقع وملاحقته.....
66	3.3.5 نتائج الدراسة Kernel approximation and Kalman Filtering.....
72	3.4 تطبيق عملي على معطيات مزود خدمة اتصالات.....
73	3.4.1 أهمية أنظمة تحديد الموقع البديلة.....
75	3.4.2 بناء نظام تحديد الموقع من معطيات شركة اتصالات.....
78	الفصل الرابع.....
78	4خوارزمية SLAM.....
79	4.1 مقدمة.....
80	4.2 لمحة تاريخية.....

80	4.3 العتاد
80	4.4 خوارزمية عمل SLAM
85	4.4.1 نقاط العلام Landmarks
85	4.4.2 ترابط المعلومات Data Association
86	4.4.3 تطبيق مرشح كالمان الاخطي في SLAM
89	الفصل الخامس
89	5 نظام وبيئة Android
90	5.1 مقدمة
90	5.2 تاريخ نظام Android
91	5.3 مميزات Android
93	5.4 بيئة تطوير Android Studio
93	5.4.1 بعض مزايا Android Studio
94	5.4.2 متطلبات تشغيل Android Studio
95	5.5 بيئة Android
96	5.5.1 مكونات التطبيق
100	5.5.2 ملف Manifest
101	الفصل السادس
101	6 نظام تشغيل الروبوت (ROS) Robot Operating System
102	6.1 مقدمة
102	6.2 تاريخ ROS وبداية تطويره
103	6.3 ميزات ROS الرئيسية وأهم نقاط قوته
105	6.4 متطلبات العمل على ROS
106	6.5 بعض المصطلحات الأساسية الواجب معرفتها في ROS

6.6 ترتيب إصدارات ROS من الأحدث للأقدم.....	108
6.7 تثبيت ROS (indigo).....	110
6.7.1 خطوات تنصيب نظام Ubuntu14.04 LTS.....	110
6.7.2 خطوات تثبيت ROS.....	111
6.8 كيف يعمل نظام ROS.....	112
6.9 الحساسات التي يتعامل معها نظام ROS.....	113
6.10 المتحكمات الممكن استخدامها مع نظام ROS.....	113
6.11 برامج المحاكاة في ROS.....	118
6.11.1 Gazebo.....	118
6.11.2 برنامج التمثيل الرسومي (RVIZ) 3D Visualization Tool.....	120
6.12 بعض التفاصيل عن عمل تطبيق Android يعمل كعقدة في نظام ROS.....	120
6.12.1 تعليمات تنصيب Android Studio.....	121
6.12.2 تعليمات تنصيب بيئة تطوير تطبيقات Android مع نظام ROS.....	121
الفصل السابع.....	122
7 روبوت Turtlebot.....	122
7.1 مقدمة.....	123
7.2 روبوت TurtleBot3 burger.....	123
7.2.1 حساس الليزر LDS-01.....	124
7.2.2 اللوحة Raspberry Pi 3 Model B.....	126
7.2.3 دائرة التحكم OpenCR.....	132
7.2.4 محركات Dynamixel.....	135
الفصل الثامن.....	138
8 بناء نظام تحديد موقع ضمن مبنى قسم هندسة الحواسيب والأتمتة في جامعة دمشق.....	138

139	8.1 مقدمة
139	8.2 مخطط العمل
140	8.2.1 تحصيل البيانات المعنونة
144	8.2.2 مرحلة بناء نظام تحديد الموقع
147	الخبرات المستفادة من المشروع
148	الآفاق المستقبلية
149	المراجع

## مقدمة

لقيت الخدمات المتعلقة بالموقع اهتماماً كبيراً في ظل التقدم التكنولوجي لأنظمة المعلومات وخصوصاً مع تطور تطبيقات الحماية والأمان وأيضاً التطبيقات التجارية وتمكن الروبوتات المستقلة من أداء المهام المختلفة معتمدةً بشكل أو بآخر على معلومات مكان وجودها.

من جانبٍ آخر، تتطلب تقنيات تحديد الموقع حسابات رياضية معقدة للحصول على أفضل تقدير ممكن للموقع. تتعدد التقنيات والطرائق الرياضية الممكن استخدامها في هذا الإطار وتتراوح بشكل عام بين الطرق التحليلية والإحصائية. عند الحديث عن طرق النمذجة الرياضية وأيضاً بعض الدارات الإلكترونية المخصصة لذلك، لا بد لنا أن ندرك أن ظهور مفاهيم تعلم الآلة والذكاء الصناعي في العقود الأخيرة أحدث أثراً كبيراً في تطور قدرة الإنسان على التنبؤ بتطور الأنظمة مع الزمن. وانعكست الثورة في هذا المجال على جوانب كثيرة جداً من جوانب الحياة.

المشروع يقدم نظرة جديدة في تطوير نظم تحديد الموقع ضمن المنشآت Indoor والأماكن الخارجية Outdoor بحيث نوّظف خوارزميات تعلم الآلة والذكاء الصناعي في استثمار المعلومات المحيطة بنا كشذات الإشارة المستقبلية RSL ومكاملتها مع الطرق الرياضية التحكمية كمرشح كالمان لنتكمن من تتبع هدف متحرك. ويعد مرشح كالمان من أهم المفاهيم الهندسية التي ظهرت في القرن الماضي وأحدثت ثورة علمية انعكس صداها في كل المجالات الهندسية. يعتبر أيضاً أداة هامة جداً في بناء أنظمة التتبع، فيكفي على سبيل المثال القول بأنه تم استخدام هذه الخوارزمية في برنامج أبولو لتتبع مسار المركبات الفضائية.



## الفصل الأول

### 1 شبكة الحساسات اللاسلكية وخوارزميات تعلم الآلة

## 1.1 مقدمة

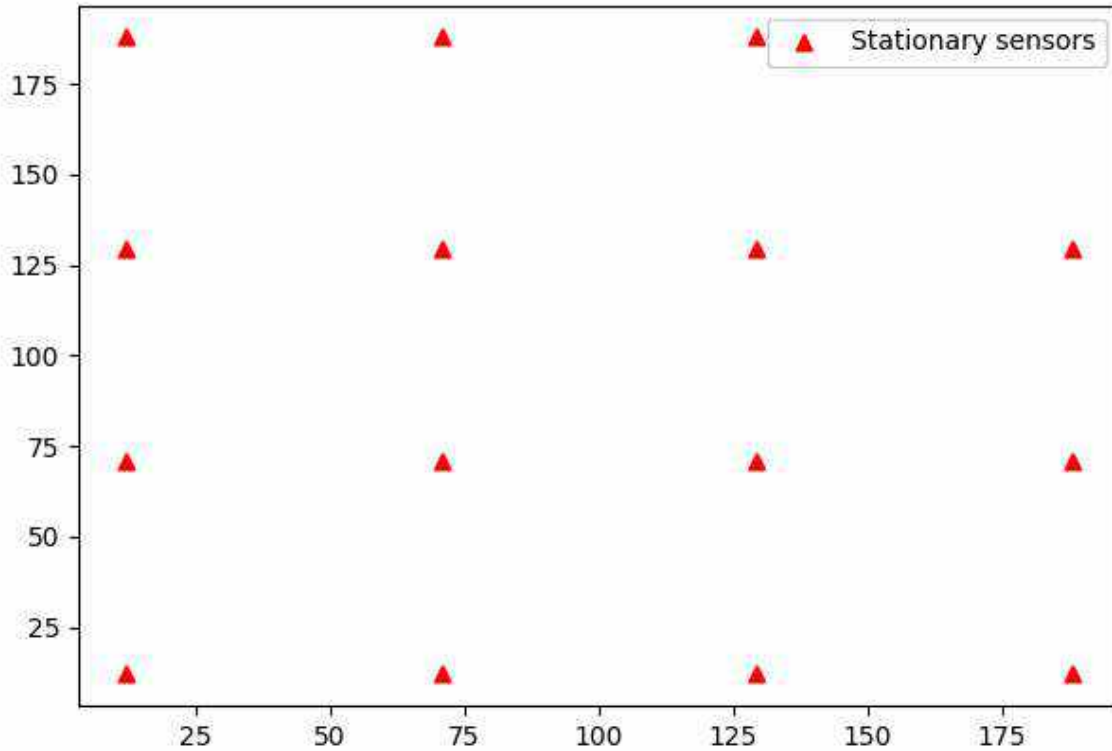
في هذا الفصل سنتعرف عن شبكات الحساسات اللاسلكية وطرق تحديد الموقع فيها إضافة إلى بعض الحديث عن خوارزميات تعلم الآلة التي سوف نستخدمها في تحديد الموقع ضمن هذه الشبكة

## 1.2 شبكة الحساسات اللاسلكية

شبكات الحساسات اللاسلكية تضم مجالاً واسعاً من التطبيقات في الرقابة والحماية وبشكل خاص تحديد الموقع ضمن الشبكة. شبكة الحساسات تحتوي على عدد كبير من النقاط التي تجعل تركيب عتاد خاص بـ GPS لكل عقدة أمر مكلف جداً فضلاً عن محدودية GPS التي تتعلق بعدم متاحته في الأماكن الداخلية Indoor وكمية الطاقة المستهلكة لتحديد الموقع. وهذا ما يقودنا إلى البحث في إمكانيات تحديد الموقع دون اللجوء إلى عتاد خاص (Thampi 2014)

### 1.2.1 بنية الشبكة اللاسلكية

تتألف البنية النظرية الشبكة اللاسلكية من عدد من الحساسات في أماكن معلومة حيث تبث إشارات راديوية في بيئتها كما في الشكل 1-1 بحيث أن كل عقدة تكون مرسلة أو مستقبلة للإشارات الراديوية أو كليهما معاً



الشكل 1—1 بنية شبكة الحساسات اللاسلكية

## 1.2.2 تحديد الموقع ضمن شبكة الحساسات اللاسلكية

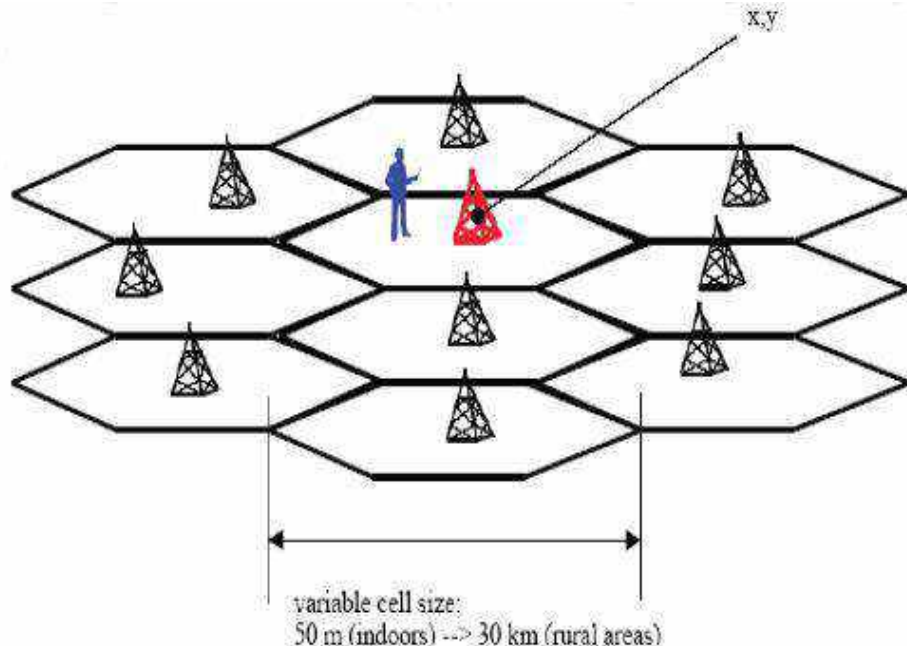
ولبناء نظام يحدد الموقع ضمنها يوجد عدة طرق منها تعتمد على بعض المتطلبات البرمجية أو العتادية أو كليهما معاً في بعض الحالات نذكر منها:

### 1.2.2.1 تحديد الموقع المعتمد على الموقع المعلوم Known Based Location Localization

وهنا التوضع يعتمد على معرفة المكان بشكل مسبق عن طريق عتاد GPS وليس لديه مرجع مكاني بالنسبة لأي عقدة أخرى. عيوب هذه الطريقة تكمن في الكلفة العالية للعتاد إضافة إلى الطاقة الكبيرة المستهلكة في تحديد الموقع وعدم متاحتها في كل الأماكن.

### 1.2.2.2 التوضع بالاعتماد على حساس القرب Proximity Based Location

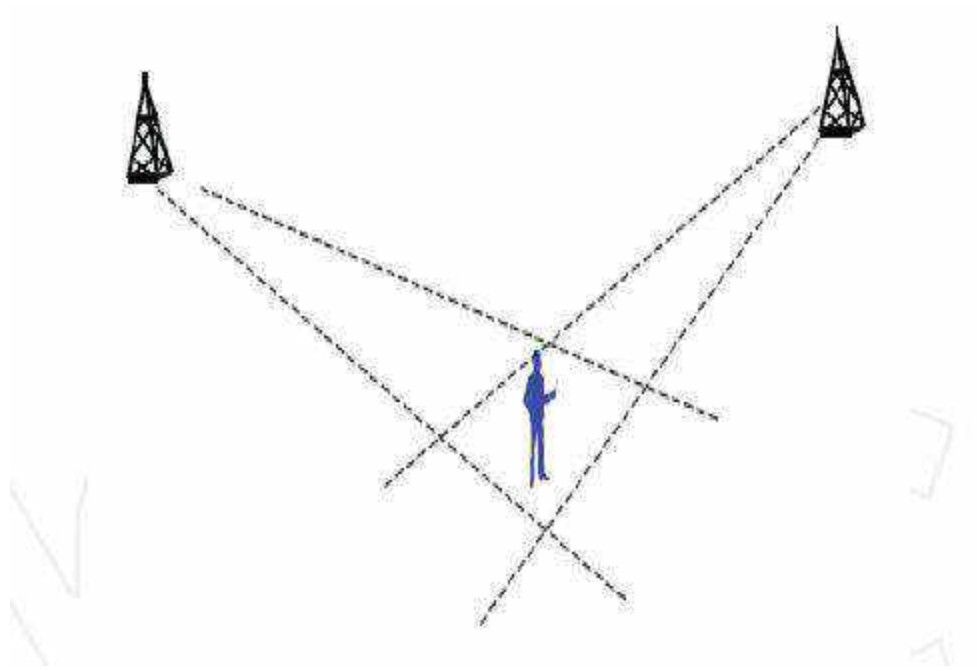
وهنا التوضع مبني على تقسيم الشبكة إلى عناقيد وكل عنقود يكون معلوم الموقع مسبقاً وباستخدام الأشعة تحت الحمراء أو البلوتوث المستخدم يوجد أقرب عنقود بالنسبة له كما في الشكل 1-2 وهذه الطريقة تشبه طريقة Cell-id المستخدمة في نظام GSM، من عيوب هذه الطريقة أنها تعتمد على اتساع مسافة العنقود فكلما كان أكبر صغرت الدقة وإضافة إلى أنه في حال تعطل أحد مراكز العناقيد يتعطل نظام تحديد الموقع في تلك المنطقة.



الشكل 1—2 تحديد التوضع حسب حساس القرب

### 1.2.2.3 تحديد الموقع المعتمد على زاوية الوصول Angle Based Localization

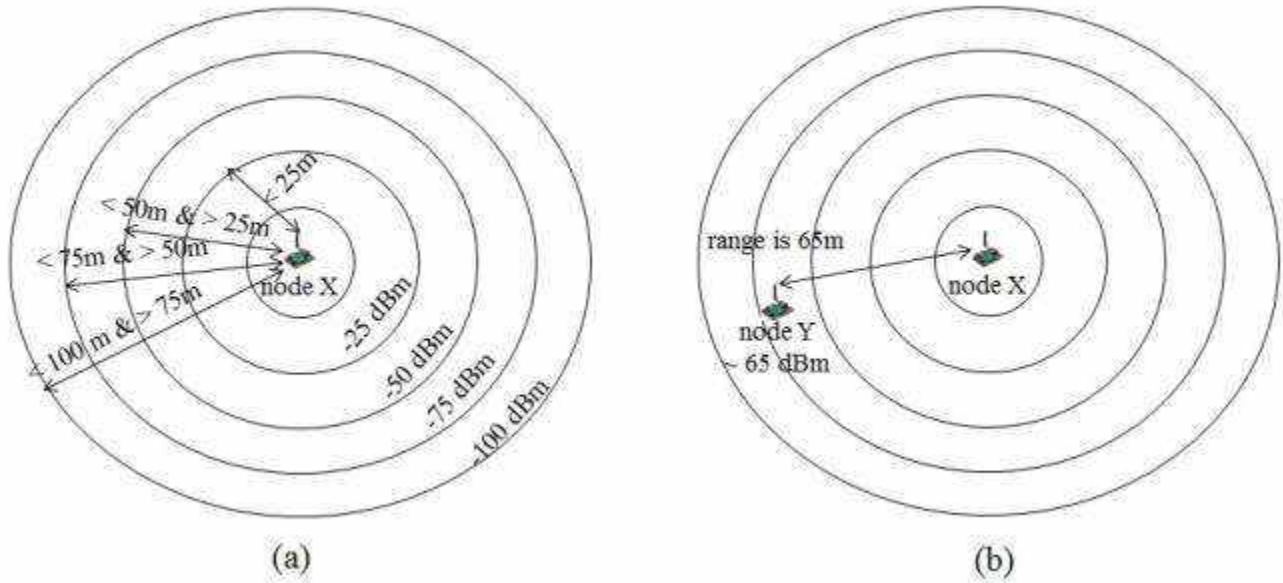
تعتمد هذه التقنية على حساب زاوية ورود الإشارة إلى المحطة القاعدية BS. تحدد هذه الزاوية خطأً خارجاً من المحطة القاعدية BS إلى المستخدم، وبذلك يلزم محطتين BS لتحديد الموقع، الذي يكون عند تقاطع الخطين الخارجين من المحطتين BSs، كما يبين الشكل 1-3 وهذه الطريقة لا تعد جيدة في المناطق المأهولة لأنها بحاجة إلى أن تكون العقدة مع المحطات على نفس خط النظر Line of sight وتحتاج إلى هوائيات خاصة ومكلفة.



الشكل 1—3 تحديد الموضع حسب AoA

#### 1.2.2.4 تحديد الموقع بالاعتماد على المسافة Range Based Localization

وهذه الطريقة تعتمد على حساب المسافة من خلال معرفة شدة الإشارة المستقبلية Received Signal Strength Indicator (RSSI) أو تعتمد على زمن الوصول Time of Arrival (ToA) أو الفرق الزمني للوصول Time Difference of Arrival (TDoA). عند استخدام طريقة شدة الإشارة RSSI فإن المرسل لديه موقع معلوم ويرسل إشارة لعقدة (مستخدم) ويتم حساب الموقع مرجعياً لهذا المرسل بالنسبة لشدة الإشارة. مشكلة هذه الطريقة تكمن في كون الإشارات عرضة للضجيج المتعلق بانتشار الأمواج الكهرومغناطيسية مما يضعف دقتها.



الشكل 1—4 المبدأ الأساسي لطريقة تحديد الموقع عن طريق شدة الإشارة المستقبلية

### 1.3 خوارزميات تعلم الآلة

يمكن، بشكل عام، تقسيم خوارزميات التعلم الآلي إلى خوارزميات خاضعة للإشراف supervised، وخوارزميات غير الخاضعة للإشراف unsupervised. يكون دخل الخوارزميات غير الخاضعة للإشراف معطيات غير معنونة ولا تحمل أي علامات مميزة مسبقاً، لتقوم عندها خوارزميات التعلم الآلي بمحاولة إيجاد أنماط متشابهة في هذه المعطيات غير المعنونة، بينما دخل الخوارزميات المشرف عليها يكون معطيات معنونة أي من أجل دخل ما نعلم ما هو الخرج الموافق له.

الخوارزميات تعلم الآلة المشرف عليها Supervised تهدف لتنبؤ قيم متحول أو عدة متحولات، بناءً على عدد من سمات المعطيات التي لها البعد  $D$ . إذا كانت قيم المتحولات المطلوب التنبؤ بها مستمرة تسمى عندها خوارزميات التعلم الآلي بخوارزميات الانحدار Regression، أما إذا كانت هذه القيم متقطعة فتسمى عندها الخوارزميات بخوارزميات التصنيف Classification.

نستعرض في الفقرات التالية الخوارزميات الخطية واللاخطية لحل مسائل الانحدار والتصنيف.

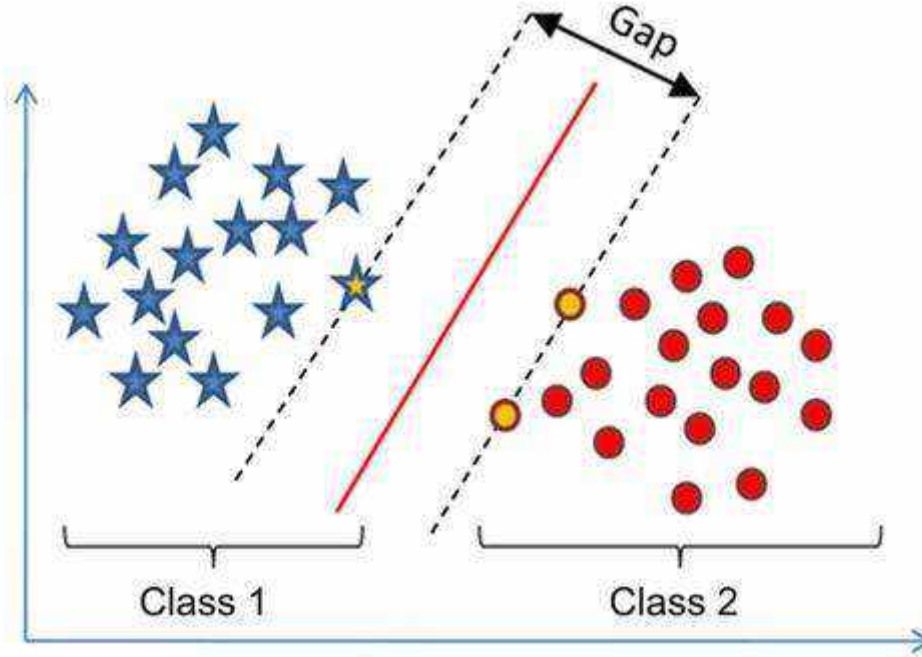
#### 1.3.1 خوارزمية Support Vector Machine

نعلم أن خوارزميات تعلم الآلة الكلاسيكية كالانحدار الخطي والشبكات العصبونية المستخدمة سواء لمسائل التصنيف أو الانحدار، يكون هدفها مقابلة الخرج  $y$  مع الدخل  $x$  هذه العملية محكومة بشعاع البارامترات  $w$  وخلال مرحلة التعلم فإن مجموعة من نقاط التدريب تستخدم للحصول على قيم لشعاع البارامترات  $w$  ومن ثم تهمل نقاط التدريب، والتنبؤ بخرج نقاط جديدة يعتمد بشكل كلي على الشعاع  $w$  (Haykin 2009)

وهناك بعض خوارزميات تعلم الآلة تأخذ مجموعة جزئية من نقاط التعليم وتستخدمها حتى ضمن مرحلة التنبؤ ومن أشهر الخوارزميات التي تعتمد هذا المبدأ هي K-Nearest Neighbor و SVM

سندرس كيفية حل خوارزمية ال SVM لمشاكل التصنيف الثنائي القابلة للفصل خطياً وهذه الدراسة تعمم كتصنيف وانحدار للنماذج اللاخطية.

بشكل أساسي SVM هو عبارة عن فاصل ثنائي مع خصائص مميزة ويتم تلخيص عمل SVM



الشكل 1-5 توضع المصنف ذو المسافة الأكبر

كما يلي: باعتبار لدينا مجموعة تدريب SVM تبني مستوي فائق كحد فاصل بين النماذج بحيث تكون مسافة الفصل margin بين النقاط الموجبة (صنف 1) والنقاط السالبة (صنف 2) أكبر ما يمكن.

### 1.3.1.1 بناء مستوي فائق أمثلي للفصل بين النماذج:

لنعتبر مجموعة التدريب حيث  $x_i$  هي نماذج الدخل للعينة رقم  $i$  و  $d_i$  هو الخرج المرغوب الموافق لتلك العينة، بفرض أن قيم  $d$  هي محدودة بـ  $1+$  ،  $1-$  ، ستكون معادلة المستوي الفائق الذي يفصل بين الصنفين :

$$w^T x_i + b = 0 \quad 1-1$$



حيث  $x$  هو شعاع الدخل و  $w$  هو شعاع البارامترات و  $b$  هو الانحياز عن المبدأ وبالتالي يمكن أن نكتب:

$$\text{for } d_i = +1 \quad w^T x_i + b \geq 0 \quad 1-2$$

$$\text{for } d_i = -1 \quad w^T x_i + b < 0 \quad 1-3$$

حيث لدينا المسافة (Margin) لشعاع الأوزان  $w$  والانحياز  $b$  هي المسافة بين المستوي الفائق وأقرب نقطة تدريب، والهدف من SVM هو إيجاد مستوي فائق بحيث تكون margin أكبر ما يمكن.

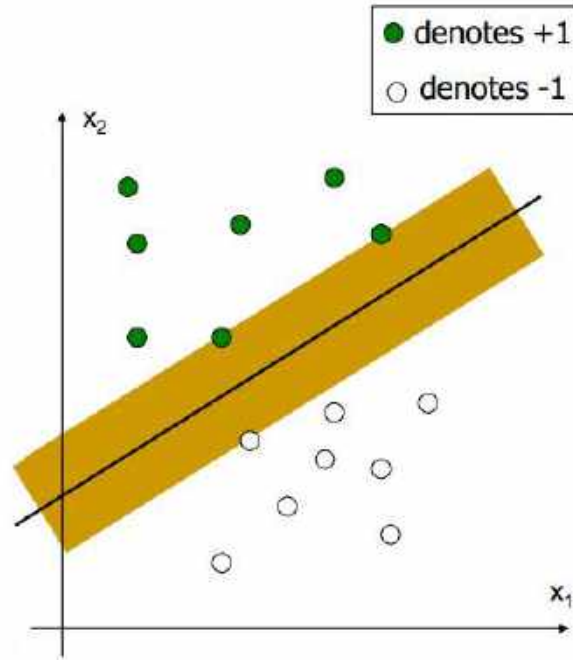
ضمن هذا الشرط الحد الفاصل يشار إليه بالمستوي الفائق الأمثلي، الشكل 1-5 يوضح كيفية تموضع ذلك المستوي ضمن فضاء دخل ثنائي البعد.

لنعتبر  $w_0$  و  $b_0$  هما القيم الأمثلية لشعاع الأوزان والانحياز على التوالي وبالتالي المستوي الفائق يمثل سطح فاصل خطي في فضاء متعدد الأبعاد:

$$w_0^T x + b_0 = 0 \quad 1-4$$

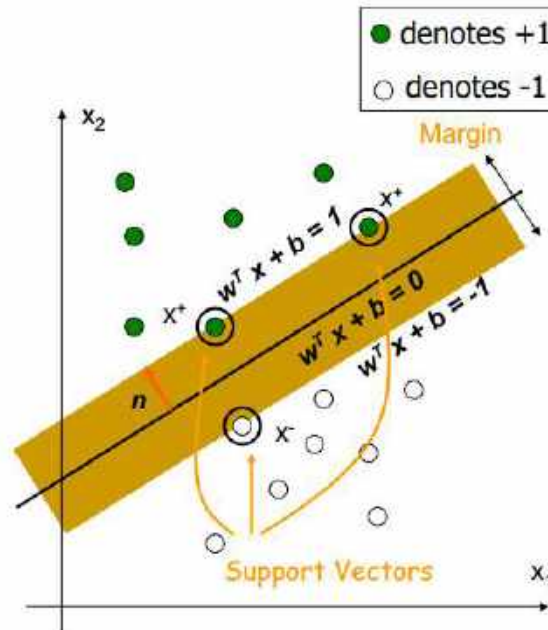
وبالتالي معادلة الدالة الفصل:

$$g(x) = w_0^T x + b_0 = 0 \quad 1-5$$



الشكل 1—6 فكرة المستوي الفائق الأمثلي

حيث لدينا المسافة من المبدأ والمستوي الفائق الأمثلي تمثل بـ  $b_0$  ومن الممكن أن تأخذ قيم موجبة أو سالبة حسب موضع المستوي بالنسبة لمبدأ الإحداثيات



الشكل 1—7 نقاط أشعة الدعم Support Vectors

الهدف هنا هو إيجاد قيم  $w_0$  و  $b_0$  للمستوي الفائق من مجموعة نقاط التدريب  $t = \{(x_i, d_i)\}$  في ضوء النتائج الظاهرة في الشكل 7-1 نرى أن  $w_0, b_0$  يجب أن يحقق الشرطين:

$$w_0^T x_i + b_0 \geq 1 \quad \text{for } d_i = +1 \quad 1-6$$

$$w_0^T x_i + b_0 < -1 \quad \text{for } d_i = -1 \quad 1-7$$

نقاط SV هي النقاط الأقرب للمستوي الفائق الأمثلي وبالتالي هي النقاط الأصعب فصلاً لأنها على تماس مباشر مع الموقع الأمثلي للحد الفاصل.

لنعتبر  $x^+$  SV الموافق للخروج  $d_i = 1$  وبالتالي من المعادلة 1-6 المسافة الجبرية بين  $x^+$  والمستوي الفائق الأمثلي:

$$w_0^T x^+ + b = 1 \quad 1-8$$

ونعبر  $x^-$  SV حيث قيمة الخروج الموافقة  $d = -1$  وتحقق المعادلة:

$$w_0^T x^- + b = -1 \quad 1-9$$

حيث الإشارة الموجبة تدل على أن  $x^+$  تقع في الجانب الموجب من المستوي الفائق الأمثلي والعكس صحيح.

القيمة الأمثلية لقيمة margin للفصل بين الصنفين المحددين بمجموعة التدريب  $t$  من المعادلة 1-8 و 1-9 تكون:

$$M = (x^+ - x^-)n$$

$$= (x^+ - x^-) \frac{w}{\|w\|} = \frac{2}{\|w\|} \quad 1-10$$

المعادلة 1-10 تعني أن تكبير المسافة margin الفصل بين الصنفين هو أيضاً تصغير طول الشعاع  $w$

أي نريد تصغير طول الشعاع  $w$  بالنسبة للقيود التالية:

*subject to:*

$$\text{for } d_i = +1, w_0^T x_i + b \geq 1 \quad 1-11$$

$$\text{for } d_i = -1, w_0^T x_i + b < 1 \quad 1-12$$

نستنتج من ذلك أن المستوي الفائق الأمثلي المعطى بالمعادلة 1-4 هو فريد حيث أن قيمة شعاع الأوزان الأمثلي  $w_0$  تعطينا الفصل الأعظمي بين النماذج السالبة والموجبة. هذا الشرط الأمثلي يحقق أيضاً تصغير طول الشعاع الأوزان  $w$ .

#### 1.3.1.1.1 حل التربيعي الأمثلي لإيجاد المستوي الفائق الأمثلي

خوارزمية SVM صيغت كي تكون من شكل المنحني المحدب الأمثلي حيث أن إيجاد حل الأمثلي للخوارزمية تتمثل بالخطوات التالية:

- إيجاد المستوي الأمثلي الفائق تبدأ أن فضاء قيم الأوزان هو مسألة أمثلية مقيدة (مشروطة)
- بناء تابع لاغرانش
- اشتقاق شروط الحل الأمثل SVM
- المرحلة الأخيرة هي إيجاد الحل الأمثلي في فضاء الثنائي لمضاريب لاغرانش

بفرض لدينا عينات تدريب

$$t = \{x_i, d_i\}_{i=1}^N \quad 1-13$$

يمكن التعبير عن الشرط بجمع المعادلتين 1-6 و 1-7 كمعادلة واحدة:

$$d_i(w^T x_i + b) \geq 1 \quad 1-14$$

$$\text{for } i = 1, 2, \dots, N$$

بهذا الشكل للقيد يمكننا بناء مسألة الحل الأمثلي المقيد كالتالي:

بوجود نماذج تدريب يجب إيجاد القيم الأمثلية لشعاع الأوزان  $w$  والانحياز  $b$  الذي يحقق الشرطين 1-11 و 1-12:

$$\text{minimize } \frac{1}{2} \|w\| \quad 1-15$$

القيد هو خطي في  $w$

$$\begin{aligned} d_i(w^T x_i + b) &\geq 1 \\ \text{for } i &= 1, 2, \dots, N \end{aligned} \quad 1-16$$

وعند نقاط SV يكون:

$$d_i(w^T x_i + b) = 1 \quad 1-17$$

لحل مثل هذا النوع من المسائل نلجأ إلى طريقة مضارب لاغرانش.

أولا نشكل دالة لاغرانش:

$$J(w, b, a) = \frac{1}{2} w^T w - \sum_{i=1}^N a_i [d_i(w^T x_i + b) - 1] \quad 1-18$$

حيث المتغيرات  $a$  تدعى مضارب لاغرانش وحل مسألة الأمثلية المقيدة هي محددة بنقطة تسمى saddle point لدالة لاغرانش  $J(w, b, a)$  التي هي حيث يوجد حلول حقيقية مترافقة (لضمان الاستقرار) وهي نقطة أصغرية بالنسبة لشعاع البارامترات  $w, b$  ولذلك للحصول على حل نقوم باشتقاق وعدم هذين الشرطين:

$$\frac{\partial J(w,b,a)}{\partial w} = 0 \quad 1-19$$

$$\frac{\partial J(w,b,a)}{\partial b} = 0 \quad 1-20$$

بتطبيق الحل الأمثلي للشرط الأول يعطينا التالي:

$$w = \sum_{i=1}^N a_i d_i x_i \quad 1-21$$

وبالحل الأمثلي للشرط الثاني يعطينا:

$$\sum_{i=1}^N a_i d_i = 0 \quad 1-22$$

نرى أن حلول شعاع  $w$  معرفة عند عدد عينات  $N$ ، والحل وحيد بما أن دالة لاغرانش دالة محدبة ولكن هذا الأمر لا ينطبق على قيم مضاريب لاغرانش.

أيضا من المهم التنويه أن قيم مضاريب لاغرانش  $a$  تأخذ قيمة (لا تساوي الصفر) فقط عند تحقق معادلة القيد 1-17:

$$a_i [d_i (w^T x_i + b) - 1] = 0 \quad 1-23$$

أي قيم  $a$  للنقاط التي ليست SV هي صفر.

المسألة الأولية 1-18 تتعامل دالة كلفة محدبة وقيود خطية، وهذه مسألة أمثلية مقيدة يمكن إنشاء منها ما يسمى المسألة الثنائية وهذه المسألة لها نفس القيمة الأمثلية للمسألة الأولية ولكن مع مضاريب لاغرانش التي تعطي الحل الأمثلي.

والآن نشكل المسألة الثنائية بتعويض قيم الحل الأمثلي 1-21 و 1-22 بالمسألة الأولية فينتج لدينا:

$$Q(a) = \sum_{i=1}^N a_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N a_i a_j d_i d_j x_i^T x_j \quad 1-24$$

بالنسبة للقيود:

$$Q(a) = \sum_{i=1}^N a_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N a_i a_j d_i d_j x_i^T x_j \quad 1-25$$

$$\sum_{i=1}^N a_i d_i = 0 \quad 1-26$$

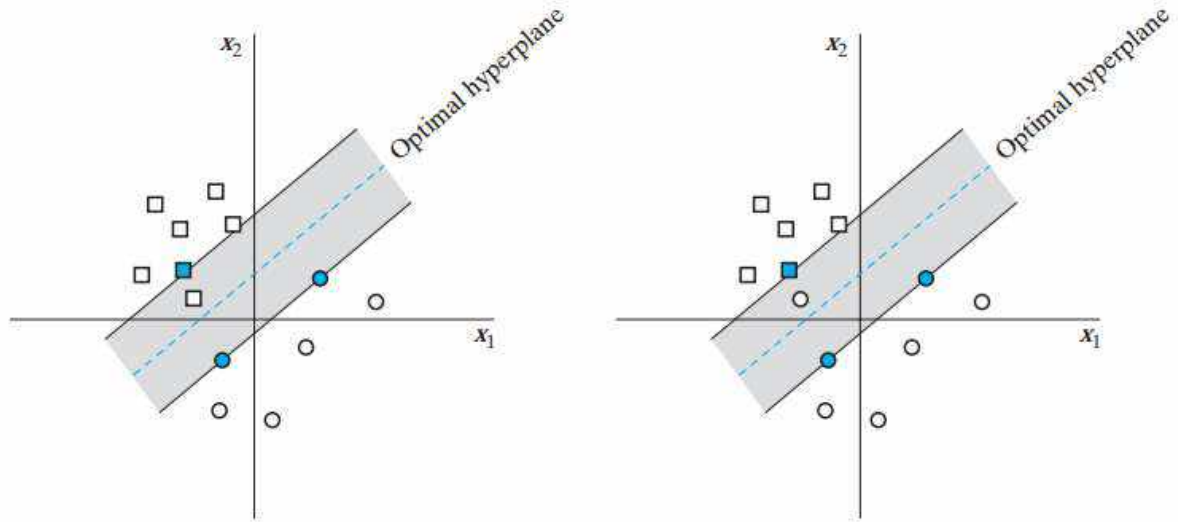
$$a_i \geq 0 \quad \text{for } i = 1, 2, \dots, N \quad 1-27$$

كما نرى أن حل المسألة الثنائية يعطى بواسطة جداء داخلي موزون لأشعة الدخل لنقاط SV

$$\{x_i^T x_j\}_{i=1}^N \quad 1-28$$

### 1.3.1.2 تعميم لحالة وجود ضجيج في عينات التدريب SVM soft margin

لقد استطعنا إنشاء مستوي فائق أمثلي في حالة نماذج التدريب كانت مثالية وخالية من حالات الضجيج ونقاط العشوائية الخاطئة التي ليس من الممكن التعامل معها بنفس النموذج السابق، والآن نرغب في بناء مستوي فائق أمثلي لعينات تدريب تحوي نماذج مصنفة بشكل خاطئ وبعض الضجيج عن طريق تخفيض احتمالية خطأ التصنيف بشكل وسطي على جميع عينات التدريب.



الشكل 1—8 نقاط الضجيج وانتهاكها لمنطقة الفصل

وقد تخل نقطة ما بالشروط:

$$d_i(w^T x_i + b) \geq 1 \quad 1-29$$

وهذا الأمر قد يحصل بإحدى هاتين الطريقتين:

- أن النقطة تقع داخل منطقة الفصل ولكن في الجانب الصحيح من الحد الفاصل كما في الجانب الأيسر من الشكل 8-1
- أن النقطة تقع في الجانب الخاطئ من الحد الفاصل كما في الجانب الأيمن من الشكل 8-1

نلاحظ في الحالة الأولى لدينا تصنيف صحيح ولكن في الحالة الثانية يوجد تصنيف خاطئ لمعالجة هذه الحالة نضيف حداً غير سالب متغير  $\{\xi_i\}_{i=1}^N$  إلى تعريف الحد الفاصل من المستوي الفائق:

$$d_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1, 2, \dots, N \quad 1-30$$



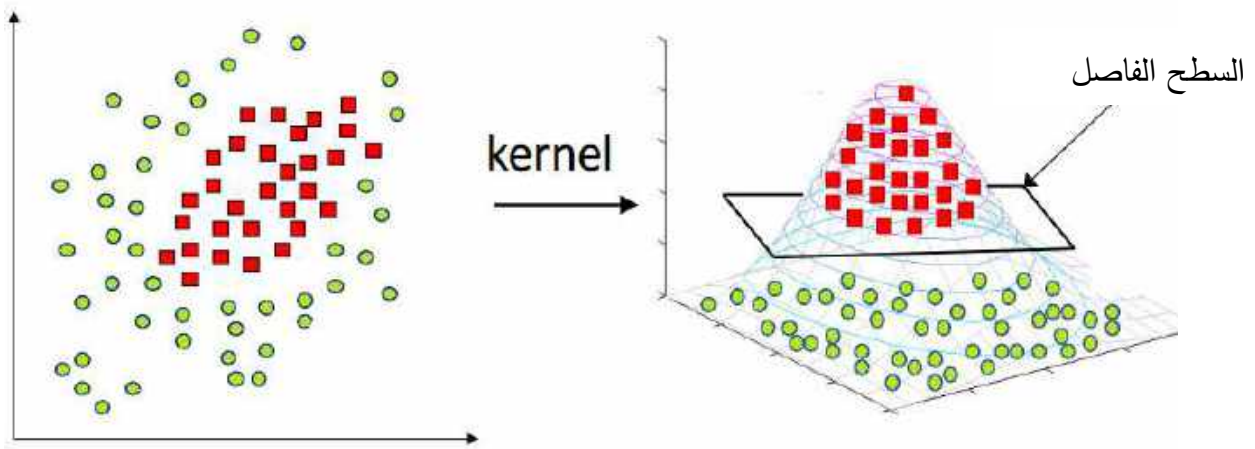
وهذا المتغير  $\xi_i$  ندعوه slack يقيس مدى خرق عينة التدريب القيد 1-16 وتكون  $0 < \xi_i \leq 1$  لعينة تقع ضمن منطقة الفصل ولكن في الجانب الصحيح من الحد الفاصل ويكون  $\xi_i > 1$  لعينة تقع في الجانب الخاطئ من منطقة الفصل وتكون قيمة  $\xi_i$  تساوي الصفر لنقاط SV وفي حال كونها لا تساوي الصفر كما في نقاط الضجيج تكون هذه النقاط غير مساهمة في بناء الحد الفاصل والآن تصبح دالة الكلفة بالشكل التالي مع إضافة عامل تقييس  $C$  الذي يتحكم بالموازنة بين تعقيد النموذج وعدد النقاط الضجيج غير قابلة للفصل:

$$\Phi(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^N \xi_i \quad 1-31$$

والمعامل  $C$  يسمى معامل التسوية regularization factor وعندما يأخذ قيمة عالية تكون الثقة كبيرة بعينات التدريب والعكس صحيح إذا كان صغيراً، تكون الثقة قليلة والبيانات تحوي ضجيج ومعرضة للخطأ.

### 1.3.1.3 تعميم من أجل النماذج غير القابلة للفصل خطياً

من إحدى أشهر الطرق استخلاص السمات المميزة في بناء نموذج تعلم الآلة هي تحويل فضاء الدخل إلى فضاء جديد يسمى فضاء السمات feature space حيث يكون هذا الفضاء أغنى بمعلومات تمكننا من تصنيف المعطيات بشكل أفضل وأحد أبسط هذه الطرق هي اخذ التربيع والتكعيب لمتحولات الدخل وإدخالها إلى خوارزمية تعلم الآلة بدلا من المتحولات الأصلية وهذا ما يسمى بال feature mapping



الشكل 1—9 تصنيف النماذج غير القابلة خطيا للفصل عن طريق Kernel

وفي SVM أيضا يمكننا استخدام متحولات بفضاء سمات  $\phi(x)$  عوضا عن استخدام  $x$  وبما أن الخوارزمية تحسب الجداء الداخلي  $\langle x_i, x_j \rangle$  هذا يعني أننا سوف نستبدل كل الجداء الداخلي  $\langle x_i, x_j \rangle$  بالجداء الداخلي  $\langle \phi(x_i), \phi(x_j) \rangle$

ويكون  $\phi$  هو feature mapping ونعرف kernel بالعلاقة:

$$k(x_i, x_j) = \phi(x_i)^T \phi(x_j) \quad 1-32$$

أي أننا لا نحتاج لمعرفة تابع تحويل السمات بشكل صريح ولكن نريد أن نحسب الجداء الداخلي لهذا التحويل وهذا أهم ميزات SVM وهذا الجداء الداخلي ل feature map يسمى kernel.

أي kernel تمكننا من التصنيف الخطي في فضاء جديد يسمى فضاء السمات لنماذج تدريب لم تكن قابلة للفصل خطيا في فضاء الدخل.

### 1.3.2 خوارزمية سمات فورييه العشوائية Random Fourier Features

تستهلك عملية تدريب الـ kernel machines الوقت بشكل كبير ولتسريع هذه العملية نقترح أن يتم تحويل نماذج الدخول إلى فضاء سمات عشوائي ومنخفض الأبعاد ومن ثم تطبيق الخوارزميات الخطية السريعة. يصمم فضاء السمات العشوائي بحيث يكون الجداء الداخلي له يساوي تقريباً الجداء الداخلي لـ kernel الأصلية. وهذا الأمر بطريقة أو بأخرى يضمن لـ SVM عدم الوقوع في مشكلة زيادة التطابق overfitting وبالتالي ممكن أن يعطينا نتائج أفضل أو تساوي النتائج المستحصلة من SVM أحياناً وربما أقل ولكن ذلك يعتبر موازنة جيدة نظراً لتسريع عملية التدريب والاختبار (Ali Rahimi 2007).

كما رأينا سابقاً SVM يعتمد على الجداء الداخلي بين نقاط التدريب وبالتالي يمكن توليد سمات عن طريق kernel trick التي تمكننا من حساب قيمة الجداء الداخلي من دون الحاجة لحساب feature map بشكل صريح أي الخوارزمية تتعامل مع نقاط التدريب عن طريق حساب قيمة kernel لها لكل نقاط التدريب وبالتالي هذا يؤدي إلى عبء حوسبي وتخزيني كبير في حال كون نقاط التدريب كبيرة العدد.

لنحل هذه المشكلة بدلا من الاعتماد الـ kernel الأصلية نقترح الاعتماد على جداء داخلي (صغير الأبعاد) عن طريق تحويل سمات عشوائي  $z$

$$k(x, y) = \langle \varphi(x), \varphi(y) \rangle \approx \langle z(x), z(y) \rangle. \quad 1-33$$

نفرض بدايةً أن الـ kernel المراد تقريبها هي shift-invariant أي تعتمد فقط على قيمة الفرق بين الشعاعين  $k(x - y)$  (Musco 2018)

نكتب  $k(x - y) = k(\Delta)$  باستخدام تحويل فورييه العكسي:

$$p(\eta) = \mathcal{F}^{-1}k(\Delta) \quad 1-34$$

$$k(\Delta) = \int p(\eta) e^{-i\pi\eta^T \Delta} d\eta \quad 1-35$$

بتقريب هذا التكامل إلى مجموع منتهي:

$$k(\Delta) \approx \frac{1}{s} \sum_{j=1}^s c_j e^{i\pi \eta_j^T \Delta} \quad 1-36$$

حيث  $c_i$  هي معاملات تؤخذ بحسب طويلة التردد  $\eta_i$   
وبما أن كل تردد يأتي بشكل مترافق فهذا التقريب يحقق ال kernel:

$$Z(x_1)Z(x_2)^* = \sum_j \frac{c_j}{s} e^{-i\pi \eta_j^T (x_1 - x_2)} \approx k(x_1 - x_2) \quad 1-37$$

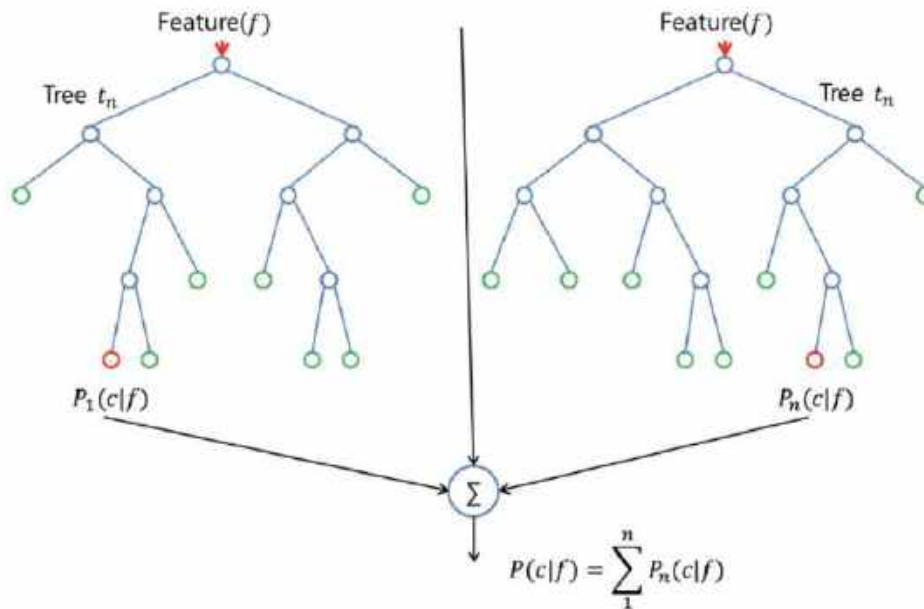
وتسمى هذه العملية ب Low-rank Kernel Approximation ويوجد عدة مقاربات لتطبيقها على غرار استثمار تحويل فورييه للkernel وبشروط مختلفة مثل طريقة Nystroem. ويعد بذلك تقريب kernel هو من أحد الطرق التي تساعدنا في عمل موازنة بين الانحياز والتباين حيث ممكن أن kernel تؤدي إلى overfitting لنموذج المراد إيجادها.

### 1.3.3 خوارزمية Random Forest

خوارزمية Random Forest هي خوارزمية مرنة وقوية في مجال تعلم الآلة حيث تعطي نتائج جيدة حتى دون ضبط معاملاتها في غالب الأحيان وهي من أكثر الخوارزميات استخداماً بسبب سهولتها وإمكانية استخدامها لمشاكل التصنيف والانحدار، في هذه الفقرة سنستعرض كيفية عمل الخوارزمية (Donges, The Random Forest Algorithm 2018)

هذه الخوارزمية تعتبر من خوارزميات تعلم الآلة المشرف عليها كما يوحي اسم الخوارزمية "الغابة العشوائية" هي تنشأ غابة وتجعلها عشوائية. الغابة التي تنشأها هي طاقم من أشجار القرار، في غالب الأحيان تدرب عن طريق Bagging (Bootstrap Agregation) الفكرة العامة لطريقة Bagging هي مكاملة عدد من نماذج التعلم لتحسين الدقة.

أحد أهم مزايا Random Forest هي إمكانية استخدامها لمشاكل التصنيف والانحدار التي تشكل معظم مسائل تعلم الآلة الحالية. الشكل 1-10 يوضح كيف يمكن أن يكون شكل Random Forest بشجرتين حيث ننفذ من نتائج الشجرة الأولى والثانية في الخرج النهائي.



الشكل 1—10 مبدأ عمل Random Forest

بارامترات Random Forest مشابهة لبارامترات أشجار القرار وأيضاً خوارزميات الـ Bagging حيث ليس هنالك حاجة لجمع شجرة قرار مع خوارزمية أخرى يتم ببساطة الجمع مع شجرة قرار أخرى.

Random Forest تضيف عشوائية للنموذج أي عندما يتم بناء الأشجار بدلاً من البحث عن السمة الأكثر أهمية لفصل العقد يبحث في فضاء جزئي عشوائي من السمات عن أفضل سمة. وهذا يعطي تنوع وتعميم أفضل للنموذج.

لهذا الأمر في خوارزمية Random Forest يؤخذ فقط فضاء جزئي من السمات بعين الاعتبار من الخوارزمية عند تقسيم عقدة. ويمكنك جعل الأشجار عشوائية أكثر عن طريق البحث عن أهمية السمة بشكل عشوائي عوضاً عن البحث عن أفضل أهمية بشكل رياضي كما تفعل أشجار القرار.

### 1.3.4 خوارزمية K-Means

كما رأينا في خوارزمية SVM حيث لدينا أزواج التدريب معنونة أي لدينا نماذج دخل  $x_i$  والخرج الموافق لها  $d_i$  وهذا يسمى بالتعليم الإشرافي Supervised Learning ولكن ماذا لو كانت لدينا المعطيات تحوي فقط على معطيات الدخل دون وجود قيمة الخرج المرغوب فيها؟ هنا نلجأ إلى ما يسمى بخوارزمية التعلم غير الإشرافي Unsupervised Learning هنا نحاول إيجاد الأنماط المتشابهة في نماذج الدخل التي تسمى العناقيد Clusters (Trevino 2016).

K-Means أحد أشهر خوارزميات Clustering حيث يكون لدينا K مركز لتوصيف العناقيد، تعتبر نقطة ما تعتبر تنتمي لعنقود في حال كون مركز العنقود أقرب مركز من جميع مراكز العناقيد الأخرى.

خوارزمية K-means وظيفتها إيجاد أفضل المراكز الأفضل بخوارزمية عودية تعمل كالآتي:

1. نسب نماذج المعطيات إلى العناقيد بحسب أقرب مركز لكل نقطة
2. تحديث قيمة مراكز العناقيد عن طريق حساب مركز ثقل النقاط المنسوبة لكل عنقود ووضع مركزه في تلك النقطة

هدف الخوارزمية هي تصغير قيمة التباين لنقاط كل عنقود وتستخدم المسافة الإقليدية كمقياس لحساب قرب نقطة ما من مركز:

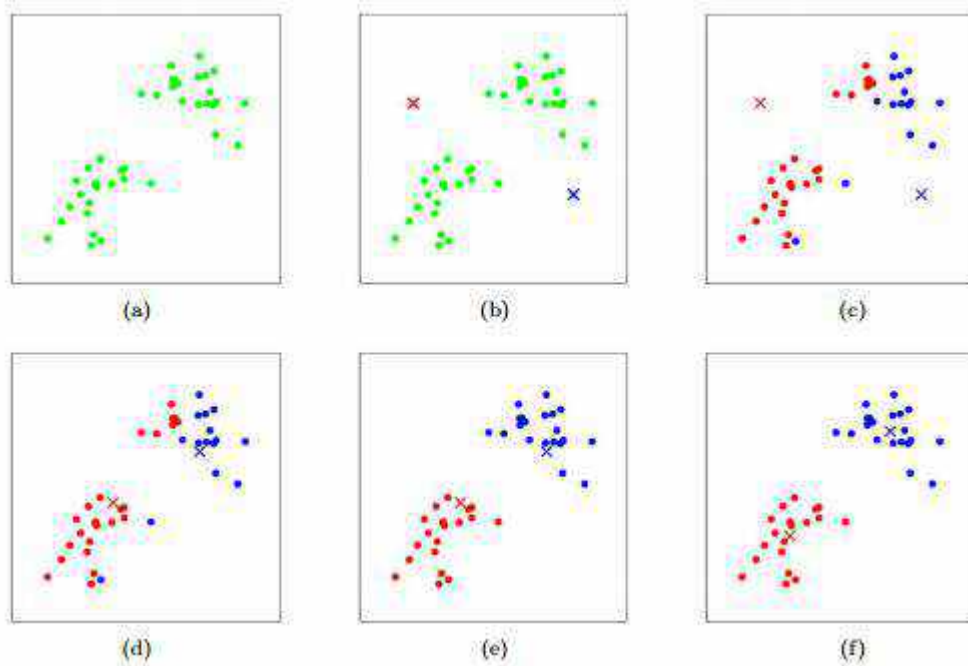
$$\min \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2 \quad 1-38$$

- دخل الخوارزمية هو عدد العناقيد المرغوب بها.
- خرج هذه الخوارزمية هو K مركز يعبر عن أفضل مراكز للعناقيد تم إيجاده

الشكل 1-11 يوضح مراحل الخوارزمية حيث نريد تجزئة المعطيات إلى عنقودين  $K = 2$  ويكون لدينا عنقود باللون الأحمر مركزه الإشارة  $x$  والمركز الثاني لونه أزرق ولديه مركز بشكل إشارة  $x$  أيضاً حيث:

1. الصورة a توضح المعطيات في البعد الثنائي

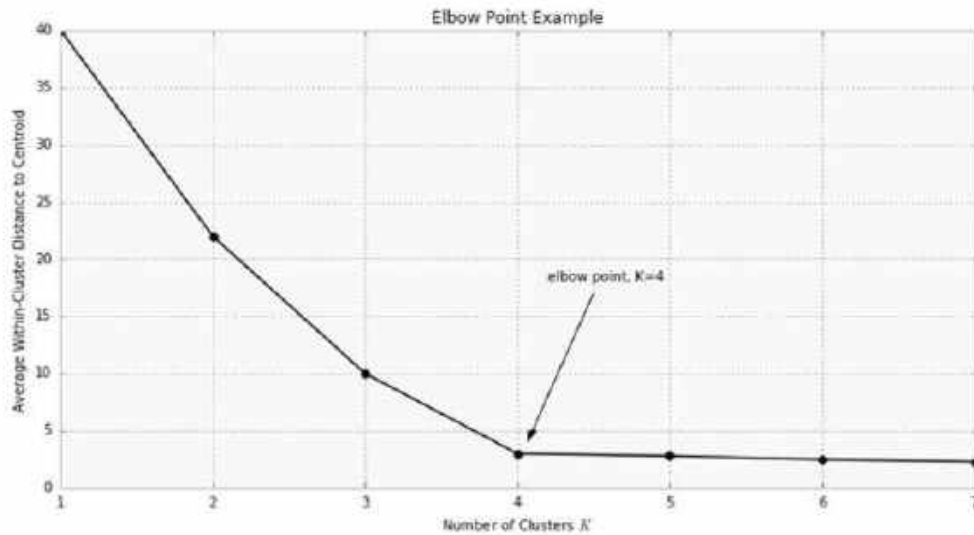
2. الصورة b توضح مواضع المراكز بشكل ابتدائي
3. الصورة c إسناد المعطيات إلى المراكز الأقرب إليهم
4. الصورة d حساب قيم المراكز الجديدة
5. الصورة e إسناد المعطيات إلى المراكز الأقرب إليهم
6. الصورة f تقارب الخوارزمية



### الشكل 1—11 مراحل خوارزمية K-Means

كما ذكرنا سابقاً دخل الخوارزمية هو عدد العناقيد المرغوب بها وفي حال كان العدد غير واضح أي لا نعلم عدد الأنماط الموجودة في المعطيات أو أنها غير محددة وهنا يمكن استخدام ما يسمى elbow-method بحيث نبحت تكرارياً بخوارزمية K-Means على قيم K ونرسم التباين عند كل قيمة كما في الشكل 1-12:





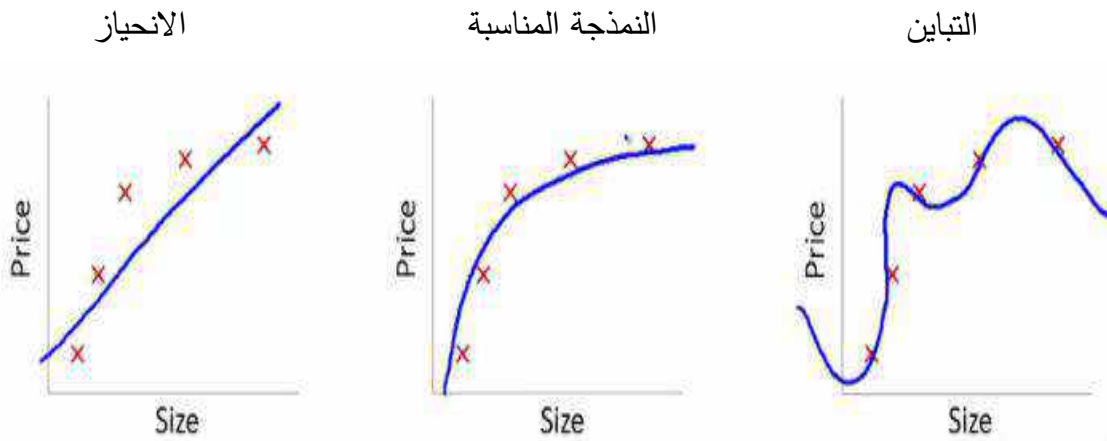
### الشكل 1—12 طريقة elbow

عمل هذه الخوارزمية يتطلب تهيئة لقيم المراكز وهناك طريقتين شهيرتين لعمل ذلك:

1. تهيئة عشوائية
2. تعيين المراكز بأحد قيم نماذج المعطيات عشوائياً

### 1.3.5 التوازن بين الانحياز والتباين

نستعرض بعضاً من الحالات المختلفة لتوابع الفرض. لنفترض أننا نريد توقع أسعار البيوت في منطقة ما بناءً على حجم هذه البيوت. للقيام بهذا الأمر نقوم بتجميع بعض المعطيات التدريبية وبناء ثلاثة توابع فرض موضحة في الشكل 1-13. يبين الرسم على اليسار تابع فرض خطي بالنسبة لمتحولات الدخل، ونستطيع أن نرى بكل وضوح بأن هذا التابع لا يلائم المعطيات التدريبية بشكل جيد، وهو بعيد عنها نوعاً ما. تسمى هذه الحالة بالانحياز. يبين الشكل على اليمين ملاءمة المعطيات التدريبية باستخدام كثير حدود، ونجد أن التابع يلائم المعطيات بشكل ممتاز، والخطأ صغير جداً. للوهلة الأولى قد يبدو لنا بأن هذا التابع سوف يعطينا قدرة ممتازة جداً على التنبؤ الأمثل. ولكن، بالأخذ بعين النظر أن المعطيات التدريبية قد تكون خاضعة لضجيج ما، نجد أن هذا التابع قد لا يكون قادراً على الخروج بتنبؤات أمثلية من أجل دخل جديد. تسمى هذه الحالة بالتباين. يوضح الرسم في المنتصف تابع فرض غير خطي، يمثل أفضل ملاءمة ممكنة للمعطيات.



الشكل 1—13 الانحياز والتباين والانحياز والنمذجة المناسبة

## الفصل الثاني

### 2 العمليات العشوائية ومرشح كالمان

في هذه الدراسة نقوم بالتقديم لمفهوم مرشح كالمان كأحد أهم مفاهيم العمليات العشوائية، وأهم المقدّرات للحالة التالية للنظام. نقوم أيضاً بالتوضيح التفصيلي لآلية عمل المرشح في توقع الحالة التالية للنظام والآلية التي يتم فيها تقليص الخطأ الناتج عن عدم التعيين المحدد بشروط المسألة. نقوم بعدها بتوضيح عمل المرشح عبر مثال تتبع حركة سيارة.

## 2.1 مقدمة

إن تطور معظم الأنظمة في الطبيعة مع الزمن يكون غير محدداً كلياً بمعادلات رياضية دقيقة، ويكون مرتبطاً بعدم تعيين ناتج عن عدة عوامل. يتمتع عدم التعيين هذا بطبيعة احتمالية مما يجعلنا ندخل عند تحليل هذه الأنظمة في مجال دراسة العمليات العشوائية Stochastic processes والتحكم بها. (Saleh 2017)

أول مشكلة يواجهها الدارس في هذا المجال هو مفهوم التقدير. حيث يتوجب تقدير الحالة التالية للنظام بناءً على الخرج المراقب وذلك لتقليص عدم التعيين المرتبط بسلوك النظام.

من أهم هذه المقدّرات (estimators) ما يعرف بـ Kalman Filter أو مرشح كالمان. مرشح كالمان هو مقدّر مثالي، يستخدم معطيات موجودة في بيئة مراقبة غير مثالية وغير دقيقة ليقوم بتوقع الحالة التالية. إن خوارزمية مرشح كالمان هي خوارزمية عودية وهذا ما يتيح استخدامها في بيانات المعالجة في الزمن الحقيقي.

يعد مرشح كالمان من أهم الخوارزميات المستخدمة لتقدير الحالة التالية للنظام ، وذلك لأنه:

- يعطي نتائج جيدة عند التطبيق العملي بسبب المثالية المتحصل عليها تحت شروط معينة للضجيج (أي أن المرشح لا يهمل وجود الضجيج بل يحاول تقليصه، وبهذا يكون سلوكه أقرب للمثالية).
- مناسب للمعالجة في الزمن الحقيقي.
- سهل الفهم والتطبيق.

يعد مرشح كالمان أداة هامة جداً في المسائل المتعلقة بتتبع مسار الأنظمة المعقدة كالصواريخ، كما يعد مفهوماً أساسياً في الملاحظة. من الجدير بالذكر أيضاً أنه تم استخدام خوارزمية مرشح كالمان لتتبع مسار رحلة أبولو.

في هذا الفصل نقوم بداية بالتقديم لمفاهيم العمليات العشوائية ومن ثم التقديم لمفاهيم خوارزمية مرشح كالمان الخطي وغير الخطي، مع إعطاء تطبيق عن تتبع حركة سيارة.

## 2.2 مفهوم العمليات العشوائية

ذكرنا سابقاً وجود العديد من الأنظمة الديناميكية المحاطة بكمية كبيرة من اللاتعيين. يؤدي هذا اللاتعيين إلى صعوبة كبيرة في صوغ نموذج رياضي قادر على التنبؤ بحالة النظام الديناميكي في المستقبل. بشكل عام يمكن القول بأن ظهور مفهوم العمليات العشوائية Stochastic Processes ناتج عن افتراض أن اللاتعيين المرتبط بالنظام هو ذو طبيعة احتمالية أو عشوائية. يمكن حقيقة وصف هذه النظم من وجهتي نظر. تتضمن وجهة النظر الأولى وصف النظام العشوائي بمعادلة فرق من الدرجة الأولى تشمل كل من حالة ودخل وخرج النظام، وبهذا ينشأ ما يعرف بسلسلة ماركوف. يمكن أيضاً وصف النظام العشوائي باستخدام معادلات فروق من درجات أعلى تشمل على دخل وخرج النظام فقط.

قبل الدخول في التوصيف الرياضي للأنظمة العشوائية نوضح عن طريق المثال التالي المفهوم العام لهذه الأنظمة.

لنفترض بأن لدينا محطة بنزين، ويراد تقدير كمية البنزين في اليوم التالي. لتقدير كمية البنزين المتوافرة يجب دراسة ثلاثة متغيرات:

1- كمية البنزين المتوافرة حتى اللحظة في محطة البنزين.

2- كمية البنزين التي يتم تعبئتها عن طريق الشاحنة.

3- كمية البنزين التي يتم صرفها في السيارات.

إن المتغير الثالث هو متغير مجهول، ذلك أننا لا نستطيع أن نعرف في هذه اللحظة كمية البنزين التي سوف تستهلكها هذه السيارات، لذلك لا يمكن تحديد كمية البنزين المتوافرة في النهار التالي بدقة. ولكن هذا المتغير ليس مجهولاً كلياً، إذ يمكن مراقبة تغيرات هذا المتغير بشكل يومي وعلى مدى طويل، وبهذا نحصل على دراسة إحصائية عن هذا المتغير.

بالإضافة لذلك فإن عملية قياس كمية البنزين المتوافرة في الكازية هي عملية تحتوي على أخطاء. بالأخذ بكل هذه الأمور بعين الاعتبار نستطيع القول بأن عملية تقدير كمية البنزين

المتوفرة بعد يوم كامل هي عملية عشوائية تحتوي على لاتعيين احتمالي يمكن دراسته إحصائياً.

يدعى الخطأ الحاصل أثناء عملية القياس بضجيج القياس Measurement Noise، أما المتغير الإحصائي الذي يدل على كمية البنزين المصروفة فيدعى بضجيج العملية Process Noise.

إذا اعتبرنا بأن كل من ضجيجي القياس والعملية هما ضجيجان عشوائيان يخضعان للتوزيع الطبيعي، يؤمن مرشح كالمان وسيلة هامة لمعرفة كمية البنزين في اليوم التالي. يعد مرشح كالمان خوارزمية عودية، ويتألف من طورين: طور التصحيح، وطور القياس. نبدأ بافتراض أولي لكمية البنزين المتوفرة في اليوم التالي بناءً على معلومات عامة وبسيطة، ومن ثم نقوم في نهاية اليوم بمقارنة كمية البنزين التي توقعناها بالكمية الحقيقية ونصحح افتراضنا وفقاً لذلك. ننتقل الآن إلى المعالجة الرياضية لكل من مفهومي الأنظمة الحتمية والعشوائية.

### 2.3 نموذج النظام الحتمي

من أجل الأنظمة الديناميكية الحتمية المتقطعة في المجال الزمني، يمكن نمذجة سلوك النظام وفقاً للمعادلة:

$$x_{k+1} = f_k(x_k, u_k), k = 0, 1, \dots \quad 1-2$$

حيث أن  $x_k \in R^n$  هي حالة النظام، و  $u_k \in R^m$  هي دخل النظام في لحظة ما  $k$ . فيما يعطى خرج النظام  $y_k \in R^p$  بالعلاقة:

$$y_k = h_k(x_k, u_k), k = 0, 1, \dots \quad 1-2$$

من أهم خصائص المعادلة 1-2 أن حالة النظام الحالية  $x_k$  وسلسلة من الدخل  $u_k, u_{k+1}, \dots, u_{k+m}$  تكفي لتحديد حالة النظام  $x_{k+m+1}$  وبشكل مستقل عن كل القيم السابقة للحالة  $x_{k-1}, x_{k-2}, \dots$  والقيم السابقة للدخل  $u_{k-1}, u_{k-2}, \dots$  أي أنه هناك تابع  $f_{k+m+1,k}$  بحيث:

$$x_{k+m+1} = f_{k+m+1}(x_k, u_k, \dots, u_{k+m}) \quad 2-3$$

## 2.4 نموذج النظام العشوائي

تعطى معادلة الحالة في النظام العشوائي بالعلاقة:

$$x_{k+1} = x_k + u_k + w_k := f(x_k, u_k, w_k) \quad 2-4$$

حيث أن  $w_k$  هو متغير عشوائي غير معروف في اللحظة  $k$ . فيما تعرف معادلة المراقبة أو معادلة القياس بالعلاقة:

$$y_k = h_k(x_k, v_k) \quad 2-5$$

حيث أن  $v_k$  هو عبارة عن متغير عشوائي غير معلوم في اللحظة  $k$ .

بالعودة إلى مثال محطة البنزين نستطيع القول بأن المتغير العشوائي  $w_k$  يمثل ضجيج العملية، أي عدم قدرة صاحب المحطة على تحديد كمية البنزين المتوفرة غداً بسبب عدم معرفته لكمية البنزين التي سيتم استهلاكها في اليوم  $k$ . أما المتغير العشوائي  $v_k$  فيمثل ضجيج القياس، أي عدم قدرة صاحب المحطة على تحديد كمية البنزين بشكل حتمي غداً بسبب عم دقة أجهزة القياس المستخدمة.

لتوصيف النظام العشوائي بشكل كامل يجب أيضاً الحصول على توزيع كل من المتغيرين العشوائيين المذكورين.

## 2.5 مرشح كالمان الخطي

نبين الآن مفهوم مرشح كالمان كمقدّر أمثلي وآلية عمله.

### 2.5.1 مفهوم مرشح كالمان وآلية عمله

يتصدى مرشح كالمان للمشكلة المعروفة بـ مشكلة تصميم المراقب Observer Design Problem. تتلخص هذه المشكلة بمحاولة تقدير الحالة الداخلية للنظام، وذلك بمعرفة خرج النظام فقط. يقوم مرشح كالمان بمعالجة هذه المشكلة بعد فرض شروط معينة للضجيج.

بكلمات أخرى، إنَّ المشكلة الأساسية التي يتصدى لها مرشح كالمان هي محاولة تقدير حالة نظام ما  $x \in R^n$  محكوم بمعادلات الفروق الخطية العشوائية التالية:

$$x_{k+1} = A_{(n \times m)} x_{k(n \times 1)} + Bu_k + w_{k(n \times 1)} \quad 2-6$$

باستخدام قياس  $z_k \in R^m$ :

$$z_{k(m \times 1)} = H_{(m \times n)} x_{k(n \times 1)} + v_{k(m \times 1)} \quad 2-7$$

حيث:

$z_k$ : الإشارة التي تعبر عن عملية القياس.

$x_k$ : الإشارة الحاملة للمعلومات، والتي تعبر عن حالة النظام.

$w_k$ : متغير عشوائي يعبر عن الضجيج المرافق للعملية، والذي له تباين Covariance معروف.

$v_k$ : متغير عشوائي يعبر عن ضجيج القياس، نعتبره ضجيجاً أبيضاً مع تباين معلوم، وغير مترابط مع الضجيج المرافق للعملية.

A: مصفوفة الانتقال من الحالة k، إلى الحالة k + 1، نفترضها ثابتة مع الزمن.

إن كلاً من المتغيرات العشوائية w و v هو ما يجعل هذا النظام نظاماً عشوائياً. فهي تضيفي عدم تعيين ضجيجي على كل من العملية الممثلة بمتحولات الحالة و متحولات القياس على الترتيب.

نعتبر كلاً من ضجيج العملية وضجيج القياس متغيرين عشوائيين مستقلين عن بعضهما، وكل منهما يخضع للتوزيع الطبيعي:

$$p(w) \sim N(0, Q) \quad 2-8$$

$$p(v) \sim N(0, R) \quad 2-9$$



حيث  $Q$  هو تغاير ضجيج العملية و  $R$  هو تغاير ضجيج القياس كما يلي:

$$\begin{cases} Q = E[w_k w_k^T] \\ R = E[v_k v_k^T] \end{cases} \quad 2-10$$

إن استخدام مصطلح مرشح، يعود لكون عملية إيجاد أفضل تقدير للحالة التالية للنظام متعلقة بترشيح الضجيج المرافق.

للحصول على مقدّر جيد للحالة التالية للنظام يجب أن يتم تحقيق شرطين:

1- القيمة المتوسطة للحالة المقدرة = القيمة المتوسطة للحالة الحقيقية.

2- الحالة المقدرة تتغير بشكل صغير قدر الإمكان عن القيمة الحقيقية.

باعتبار أن كل من الضجيج المرافق للحالة المراد تقديرها والضجيج المرافق لعملية القياس هما ضجيج أبيض ذو توزيع طبيعي، يقوم مرشح كالمان بتقدير الحالة التالية للنظام بشكل أمثلي، لأنه يصغر الخطأ التربيعي المتوسط وذلك عن طريق تصغير أثر مصفوفة تغاير الخطأ.

يمكن تقسيم عمل المرشح إلى طورين: طول التنبؤ وطور التصحيح.

طور التنبؤ: في هذا الطور يتم توقع الحالة التالية للنظام باستخدام معادلات الحالة التي توصف النظام والخطأ المقدّر. تعطى معادلات طور التنبؤ كما يلي:

$$\hat{x}_k' = Ax_{k-1} \quad 2-11$$

$$P_k' = AP_{k-1}A^T + Q \quad 2-12$$

حيث أن المصفوفة  $A$  هي المصفوفة التي تحدد تغير حالة النظام بين نقاط زمنية متعاقبة. أما  $P_k$  فتتمثل تغاير التقدير التالي.

طور التصحيح: في هذا الطور يتم تطبيق آلية تغذية خلفية. نحصل بداية على قياس لخرج النظام، ومن ثم نقوم باستخدام تغذية خلفية مع ربح يسمى بربح كالمان. عن طريق تعبير ربح

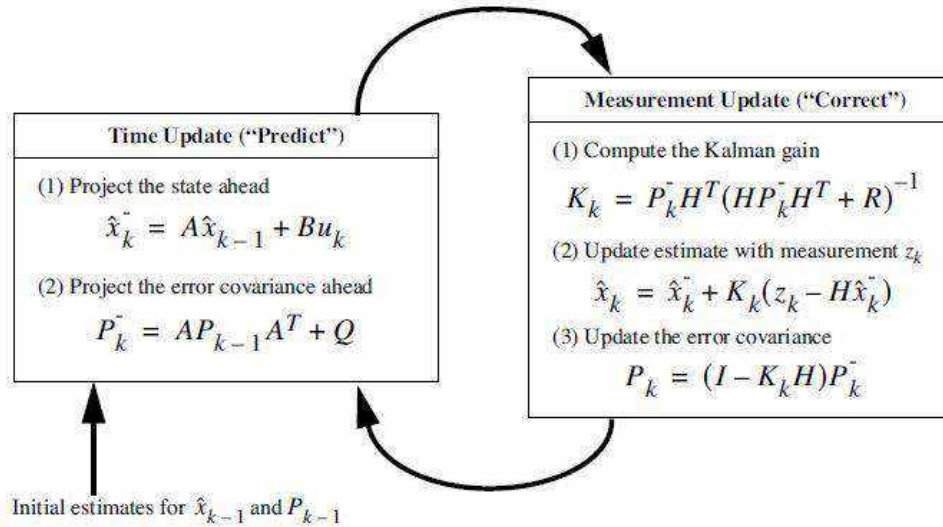
كالمان، نسعى لجعل الخطأ المرتكب أقل ما يمكن. يتم استخدام عملية القياس هذه للحصول على توقع أفضل للمرحلة القادمة. تعطى معادلات طور التصحيح كما يلي:

$$\hat{x}_k = \hat{x}'_k + K_k(z_k - H\hat{x}'_k) \quad 2-13$$

$$P_k = (I - K_k H)P'_k \quad 2-14$$

$$K_k = P'_k H^T (H P'_k H^T + R)^{-1} \quad 2-15$$

حيث أن  $K_k$  هو ربح كالمان. يتعاقب الطوران ويتلو الواحد الآخر في عملية دورية كما هو موضح في الشكل 1-2 ولكن هذا غير ضروري في كل الأحوال، إذا كان القياس غير متوفراً لبعض الأسباب يمكن القيام بعدة أطوار متعاقبة من التنبؤ. وبالمثل، إذا كانت عدة قياسات متوفرة في نفس الفترة الزمنية، يمكن القيام بعدة عمليات تصحيح في نفس الوقت.



الشكل 1—2 الطبيعة العودية لمرشح كالمان

## 2.6 مرشح كالمان غير الخطي (EKF)

في كثير من التطبيقات العملية تكون العملية التي نريد تقديرها والتحكم بها عملية غير خطية، لا يمكن التعامل معها وفق الخوارزمية الموصوفة أعلاه. لذلك ظهرت خوارزمية مرشح كالمان غير الخطي أو مرشح كالمان الممدد EKF لتحل هذه المشكلة. إن خوارزمية مرشح كالمان غير الخطي مبنية على تقريب العملية تقريباً خطياً حول التقدير السابق وفقاً لمنشور تايلور.

يعاني مرشح كالمان غير الخطي أحياناً من بعض المشاكل، فتقارب التقدير قد يكون غير دقيقاً إذا كانت الشروط الابتدائية (أو التقديرات الابتدائية) غير دقيقة، أو إذا كانت التقريب الخطي لا يصف النظام بشكل دقيق. في هذه الحالات يفضل استخدام إحدى الطرق المبنية على خوارزميات مونت كارلو، مثل المرشح الجزيئي.

بشكل مشابه نعرف حالة نظام ما غير خطي ونعرف معادلة الفروض المعبرة عن القياس كما يلي:

$$x_k = f(x_{k-1}, u_k, w_{k-1}) \quad 2-16$$

$$z_k = h(x_k, v_k) \quad 2-17$$

حيث أن كلا من  $h$  و  $f$  توابع غير خطية. من المهم هنا ملاحظة أن كلا من ضجيج العملية وضجيج القياس في هذه الحالة لم يعودا يخضعان للتوزيع الطبيعي، وذلك بنتاج التابع غير الخطي.

بإهمال كل من ضجيجي العملية والقياس نستطيع الحصول على التقريبات التالية:

$$\tilde{x}_k = f(\widehat{x}_{k-1}, u_k, 0) \quad 2-18$$

$$\tilde{z}_k = h(\widehat{x}_k, 0) \quad 2-19$$

حيث أن:

$\widehat{x}_{k-1}$ : تقدير الحالة التالي posteriori estimation الناتج عن الخطوة السابقة (k - 1).

نكتب الآن معادلات حالة جديدة تمثل تقريباً خطياً للنظام حول المعادلتين 2-18 و 2-19 :

$$x_k = \tilde{x}_k + A(x_{k-1} - \widehat{x_{k-1}}) + W w_{k-1} \quad 2-20$$

$$z_k = \tilde{z}_k + H(x_{k-1} - \widehat{x_{k-1}}) + V v_k \quad 2-21$$

حيث:

$x_k, z_k$ : القيم الحقيقية لتمثيل الحالة.

$\tilde{x}_k, \tilde{z}_k$ : القيم التقريبية الناتجة عن إهمال ضجيجي العملية والقياس.

$A, H, W, V$ : مصفوفات الجاكوبي التي تعبر عن الحد الأول من منشور تايلور حول التقدير السابق للحالة والدخل:

$$A_{[i,j]} = \frac{dF_{[i]}(\widehat{x_{k-1}}, u_k, 0)}{dx_{[j]}} \quad H_{[i,j]} = \frac{dh_{[i]}(\widehat{x_{k-1}}, u_k, 0)}{dx_{[j]}} \quad 2-22$$

$$W_{[i,j]} = \frac{dF_{[i]}(\widehat{x_{k-1}}, u_k, 0)}{dw_{[j]}} \quad V_{[i,j]} = \frac{dh_{[i]}(\widehat{x_{k-1}}, u_k, 0)}{dv_{[j]}} \quad 2-23$$

نعرف الآن الخطأ الناتج عن التقريب الذي قمنا به:

$$e_{\tilde{x}_k} = x_k - \tilde{x}_k \quad 2-24$$

$$e_{\tilde{z}_k} = z_k - \tilde{z}_k \quad 2-25$$

بشكل مشابه لمعادلات الحالة المعرفة في 2-6 و 2-7 وبلاستعانة ب 2-24 و 2-25

نعرف خطأ العملية error process كما يلي:

$$e_{\tilde{x}_k} = A(x_{k-1} - \widehat{x_{k-1}}) + \gamma_k \quad 2-26$$

$$e_{\tilde{z}_k} = H e_{\tilde{x}_k} + \beta_x \quad 2-27$$

حيث:

$\gamma_k, \beta_k$ : متغيرات عشوائية جديدة، مركزها الصفر، ولها تغاير  $Q$ ، و  $R$  المعرفان في 10 – 2، كما لهما مصفوفات التغاير:  $WQW^T, VRV^T$ .

بشكل عام، تعرف هذه المتغيرات العشوائية كما يلي:

$$P(e_{\tilde{x}_k}) \sim N(0, E(e_{\tilde{x}_k} e_{\tilde{x}_k}^T)) \quad 2-28$$

$$P(\gamma_k) \sim N(0, WQ_kQ^T) \quad 2-29$$

$$P(\beta) \sim N(0, VR_kR^T) \quad 2-30$$

باستخدام العلاقة 25 – 2 وبافتراض أننا نستخدم مرشح كالمان آخر افتراضي، نعرف تقديرنا لخطأ التقريب المعرف في 26 – 2 بالرمز  $\hat{e}_k$ . باستخدام هذا التقدير مع المعادلة 25 – 2 للحصول على تقديرات حالة تالية للعملية الأساسية نكتب:

$$\hat{x}_k = \tilde{x}_k + \hat{e}_k \quad 2-31$$

إن معادلة كالمان المستخدمة لتقدير  $\hat{e}_k$  هي:

$$\hat{e}_k = K_k \cdot e_{z_k} \quad 2-32$$

بتعويض 32 – 2 في 31 – 2 وبالاستفادة من 25 – 2 نرى أننا لا نحتاج حقيقة لاستخدام مرشح كالمان ثاني افتراضي:

$$\hat{x}_k = \tilde{x}_k + K_k e_{z_k} = \tilde{x}_k + K_k (z_k - \tilde{z}_k) \quad 2-33$$

حيث  $\tilde{x}_k, \tilde{z}_k$  معرفة في 2-18 و 2-19.

نعرف الآن المعادلات الحاكمة لمرشح كالمان غير الخطي.

معادلات طور التنبؤ:

$$\hat{x}_k' = f(x_{k-1}, u_k, 0) \quad 2-34$$

$$P_k' = A_k P_{k-1} A_k^T + w_k Q_{k-1} w_k^T \quad 2-35$$

معادلات طور التصحيح:

$$\hat{x}_k = \hat{x}_k' + K_k(z_k - h(\hat{x}_k', 0)) \quad 2-36$$

$$P_k = (I - K_k H) P_k' \quad 2-37$$

$$K_k = P_k' H_k^T (H_k P_k' H_k^T + V_k R_k V_k^T)^{-1} \quad 2-38$$

## 2.7 مثال عددي

لنفترض الآن أننا نريد أن نمذج سيارة تتحرك على طريق مستقيم. نعتبر أن حالة النظام هي عبارة عن موقع السيارة وسرعتها. دخل النظام هو التسارع المطلوب، والخرج المقاس هو موضع السيارة. لنقل بأننا نستطيع أن نغير من التسارع المطلوب والموقع كل  $T=0.5$  ثانية. بالاعتماد على معادلات الحركة نكتب تمثيل الحالة للنظام:

$$x_k = \begin{pmatrix} p_k \\ v_k \end{pmatrix} = A x_k + B u_k + w_k = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} x_k + \begin{pmatrix} \frac{T^2}{2} \\ T \end{pmatrix} u_k + w_k \quad 2-39$$

حيث  $w_k$  يعبر عن الضجيج الحاصل عن كل من الخطأ الحاصل في حساب الموضع والسرعة. أما تمثيل الحالة الموافق للخروج المراقب:

$$y_k = Hx_k + v_k = [1 \ 0]x_k + v_k \quad 2-40$$

حيث  $v_k$  ضجيج القياس.

نفترض أن القيمة المتوسطة لكل من  $w$  و  $v$  هي صفر. نفترض أيضاً أن هذين المتغيرين العشوائيين مستقلان عن بعضهما. نكتب الآن معادلات مرشح كالمان:

$$\hat{x}_k = (\hat{x}'_k + B u_k) + K_k(z_k - H\hat{x}'_k)$$

$$P_k = (I - K_k H)P_k'$$

$$K_k = P_k' H^T (H P_k' H^T + R)^{-1}$$

نعرف الآن:

$$S_w = E[w w_k^T] : \text{تغاير ضجيج العملية.}$$

$$S_v = E[v v_k^T] : \text{تغاير ضجيج القياس.}$$

نلاحظ أنه إذا كان ضجيج القياس كبيراً، يكون  $S_z$  كبير، و  $K$  صغير، أي أننا لا نعطي مصداقية كبيرة للقياس، والعكس بالعكس.

بفرض أن:

- قياس الموقع يتم بخطأ 10 أقدام وبذلك يكون:  $S_z = 100$
- التسارع المطبق ثابت وقيمه:  $1 \text{ foot/sec}^2$
- ضجيج التسارع:  $0.2 \text{ feet/s}^2$
- يتم قياس الموضع عشر مرات في الثانية، وبذلك:  $T=0.1 \text{ sec}$

نعوض في 2-39 و 2-40 :

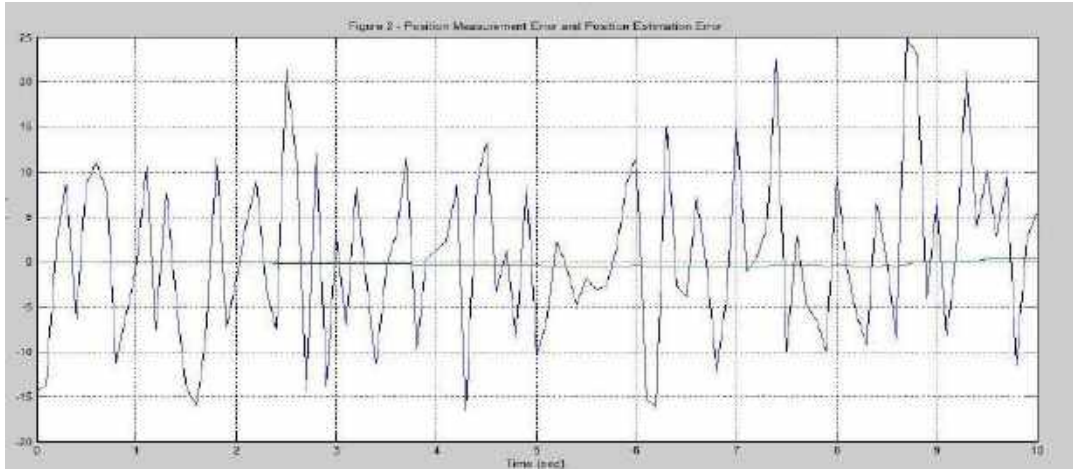
$$x_k = \begin{pmatrix} p_k \\ v_k \end{pmatrix} = Ax_k + Bu_k + w_k = \begin{pmatrix} 1 & 0.1 \\ 0 & 1 \end{pmatrix} x_k + \begin{pmatrix} 0.005 \\ 0.1 \end{pmatrix} u_k + w_k \quad 2-41$$

$$y_k = [1 \ 0]x_k + v_k \quad 2-42$$

حساب  $S_w$ :

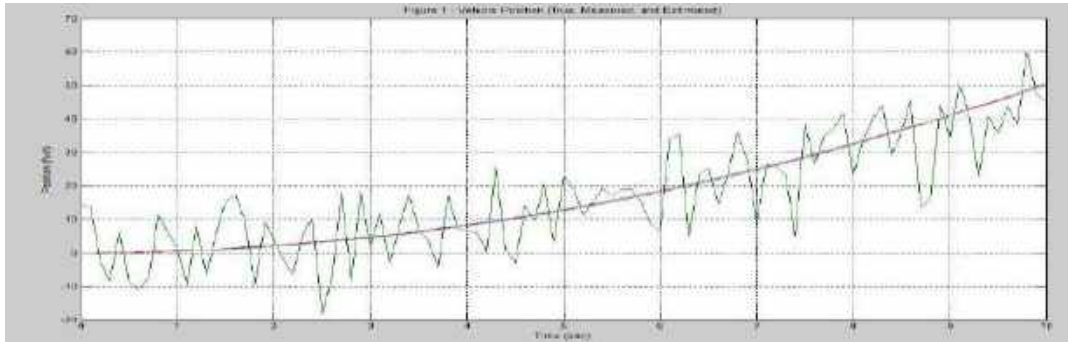
$$S_w = E(x x^T) = E \left( \begin{bmatrix} P \\ V \end{bmatrix} [P \ V] \right) = \begin{pmatrix} P^2 & PV \\ VP & V^2 \end{pmatrix} = \begin{pmatrix} 10^{-6} & 2 \times 10^{-5} \\ 2 \times 10^{-5} & 4 \times 10^{-4} \end{pmatrix} \quad 2-43$$

باعتبار  $\hat{x}_0$  أفضل تقدير للحالة الابتدائية،  $P_0 = S_w$  عدم التعيين في التقدير الابتدائي، نحصل على النتائج التالية:

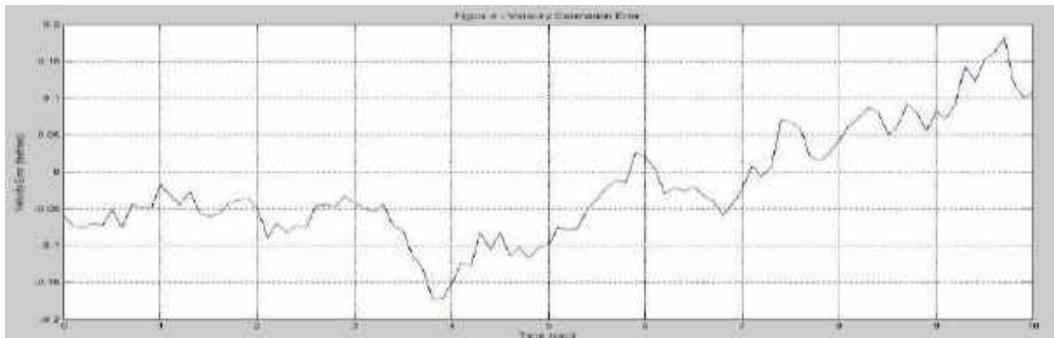


الشكل 2—2 مقدار الخطأ في تقدير الموقع لكل من عملية القياس (باللون الأزرق)، وعملية القياس بعد استخدام مرشح كالمان (باللون الأخضر).

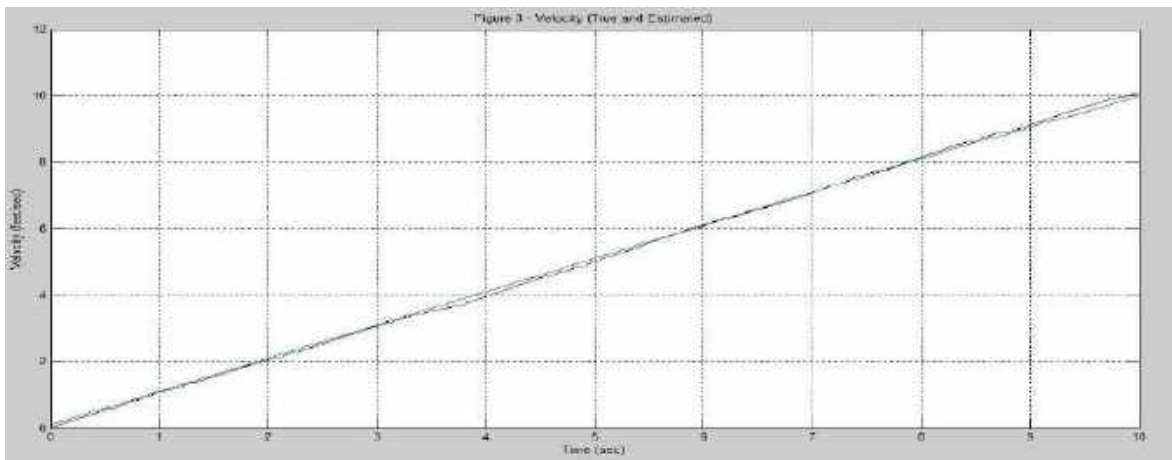




الشكل 2—3 موضع السيارة الحقيقي، المقاس، والنتائج عن مرشح كالمان. يظهر اللون الأخضر في الصورة التالية القيم المقاسة لموضع السيارة، فيما يظهر اللونان الأزرق والأحمر القيم الحقيقية والمتوقعة.



الشكل 2—4 مقدار الخطأ في قياس السرعة



الشكل 2—5 القيم الحقيقية للسرعة والقيم المتنبئ بها عن طريق مرشح كالمان

### الفصل الثالث

3 الدراسة الأمثلية لتحديد الموقع وتطبيق عملي على معطيات شركة الاتصالات سيريتل (Syriatel)

## 3.1 مقدمة

في هذا الفصل سنستعرض كيفية استخدام طريقة kernel approximation في شبكة الحساسات اللاسلكية للحصول على نتائج أمثلية ومن ثم إنشاء نظام تحديد الموقع ببيئة خارجية Outdoor بالاعتماد على معطيات عن إشارات الشبكة الخلوية الموافقة لموقع المستخدم تم تقديمها من مزود خدمة الاتصالات سيريتل.

## 3.2 التعريف بأهم المكتبات المستخدمة

في جميع التطبيقات اللاحقة تم العمل على لغة Python وهي من لغات البرمجة عالية المستوى وستعرض أهم المكتبات التي تم استخدامها:

- **مكتبة Numpy**

وهي المكتبة الأساسية للحسابات العلمية في بايثون وتزودنا بمصفوفات متعددة الأبعاد عالية الأداء وأدوات للتعامل مع هذه المصفوفات وهي النسخة المحسنة من مكتبة Numeric.

Travis Oliphant هو من إنشائها عن طريق دمج ميزات مكتبة Numarray المنافسة مع مكتبة Numpy مع تعديلات كثيفة.

- **مكتبة Scikit-learn**

هي برمجيات مجانية مخصصة لمجال تعلم الآلة وتعد من أشهر مكتبات علم البيانات على الإطلاق، مشروع scikit-learn بدأ في البرنامج السنوي Google Summer of Code من قبل David Cournapeau وفي 2010 تم إعادة إنشائها من قبل مطورين آخرين نظرا للاهتمام الكبير بها، وتحتوي على العديد من الميزات مثل تطبيق خوارزميات الانحدار والتصنيف المختلفة وغيرها (Fabian Pedregosa 2011)

- **مكتبة Pandas**

دائما ما كانت بايثون جيدة لأجراء التحويلات على المعطيات ولكنها ليست جيدة لتحليل المعطيات بقدر استخدام SQL أو Excel أو R. Pandas هي مكتبة مفتوحة المصدر تزودنا ببنى معطيات سريعة ومرنة وقد صممت بحيث يمكن التعامل مع الجداول العلائقية بشكل فعال وقد طُورت من قبل

Wes McKinney في 2008 حيث تعد Pandal مناسبة للعمل مع جداول المعطيات المتسقة حسب العمود كما في جداول Excel,SQL

#### • مكتبة Matplotlib

هذه المكتبة هي إضافة رياضية عددية لـ Numpy و مكتبة رائعة لعرض أو إظهار المعطيات وهي تمكن المحللين وأصحاب القرار من رؤية التحليلات على هيئة صور كي يتمكنوا من استيعاب المفاهيم الصعبة أو يتعرفوا على النماذج الجديدة.

### 3.3 دراسة الأمثلية لخوارزميات تعلم الآلة ضمن شبكة الحساسات اللاسلكية

في ظل التقدم الكبير والاهتمام في مجال تعلّم الآلة والذكاء الصناعي دائما يوجد خيارات وخوارزميات كثيرة يمكن استخدامها لحل مشكلة ما وليس هنالك حل محدد أو منهج ثابت يمكن اتباعه لحل جميع المشاكل وإنما علينا الاعتماد على الحس الهندسي لاستنتاج الحلول الأفضل. في هذا الفصل سنستعرض نتائج استخدام تقنيات Kernel approximation مع SVM لتنبؤ بالموقع بالإضافة إلى مرشح كالمان لملاحقة هدف ضمن شبكة حساسات لاسلكية وأيضاً دراسة عملية مبنية على معطيات من مزود خدمة اتصالات لإمكانية تعميم مبدأ شبكة الحساسات اللاسلكية وتموضعها على شبكة الاتصالات خلوية.

#### 3.3.1 منهجية العمل

سنعتمد في تحديد الموقع على طريقة RSL إذ تعتبر دقة هذه الطريقة جيدة نسبياً تعتمد تقنية RSL على أنّ شدة الإشارة المستقبلية من محطة قاعدية ما تتناسب مع البعد عن هذه المحطة، لذلك، ويتوافر شدة الإشارة المستقبلية من ثلاث محطات قاعدية يمكن، نظرياً، تحديد الموقع المستخدم عبر استخدام مفهوم التثليث. ولكنّ المشكلة حقيقةً تكمن في صعوبة الحصول على علاقة تحليلية تربط بين شدة الإشارة المستقبلية والمسافة عن المحطة القاعدية، حيث أن ظروف الانتشار وطبيعة وسط الانتشار يختلفان بشكل كبير من منطقة لأخرى. لاستخدام هذه التقنية في تحديد الموقع لا بدّ أولاً من تحديد علاقة رياضية تربط بين شدة الإشارة المستقبلية والمسافة عن المحطات القاعدية.

تختلف الدراسات فيما بينها في بناء هذه العلاقة الرياضية. يمكن بشكل عام تقسيم هذه الطرائق إلى:

- **طرائق تحليلية:**

تعتمد هذه الطرائق على بناء علاقات رياضية تحليلية تجريبية تربط بين شدة الإشارة المستقبلية والمسافة عن المحطة القاعدية.

- **طرائق البصمة الراديوية:**

تعتمد هذه الطرائق على بناء نماذج رياضية خاصة بكل منطقة على حدى. فمثلاً تختلف شروط انتشار الإشارة بشكل كبير بين المناطق الريفية والمناطق المدنية. في هذه التقنيات يتم بناء النموذج الرياضي بناءً على عدد كبير من القياسات. نستطيع أن نميز من هذه الطرق:

- **طرائق تعتمد على المقارنات:**

في هذه الطريقة يتم تجميع عدد كبير من المعطيات، تمثل شدات الإشارة المستقبلية من عدة محطات قاعدية وفي عدة مناطق، والموقع الموافق لها. بعد تجميع عدد كافٍ من المعطيات، يصبح النموذج جاهزاً للاستخدام. من أجل دخل جديد، يتم مقارنة شدات الإشارة المستقبلية من عدة محطات قاعدية مع الجدول الموجود، ومن ثم يتم اختيار الموقع الموافق. تعاني هذه الطريقة من الدقة الضعيفة ومن التكلفة الحسابية العالية لحساب الموقع من أجل دخل جديد، لأن عمليات المقارنة مكلفة حسابياً.

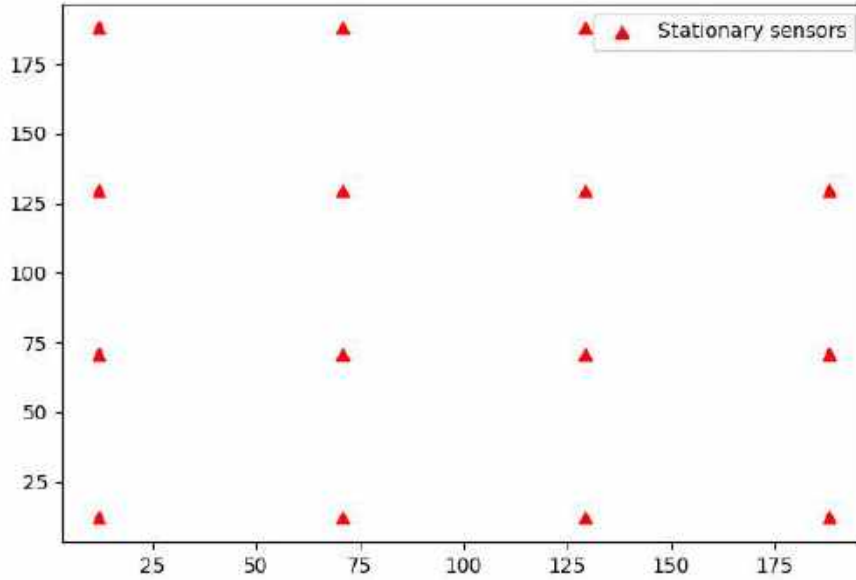
- **طرائق تعتمد على خوارزميات التعلم الآلي:**

في هذه الطرائق يتم أيضاً تجميع عدد كبير من المعطيات، تمثل شدات الإشارة المستقبلية من عدد من المحطات القاعدية والمواقع الموافقة، ومن ثم يتم بناء تابع يربط شدات الإشارة بالموقع عبر عملية التدريب. بعد انتهاء التدريب يصبح من الممكن استخدام هذا التابع للتنبؤ بالموقع الموافق لدخل جديد. تستخدم الدراسة (Sandy Mahfouz 2014) هذه الطريقة لتحديد الموقع في شبكات حساسات لاسلكية.

نعتمد في مشروعنا على بناء النموذج الرياضي بالاعتماد على خوارزميات التعلم الآلي، لأنها تعطي أدق نتائج ممكنة.

### 3.3.2 نموذج أوكومورا-هاتا Okumura-Hata Model

بافتراض بيئة ذات مساحة معينة ويوجد فيها عدد من الحساسات كيف من الممكن محاكاة شدة الإشارة المستقبلية عند موقع ما ضمن الشبكة؟ هنا لجأنا إلى نموذج Okumura-Hata



الشكل 3—1 أبعاد البيئة وأماكن تموضع الحساسات

لكافة النتائج حول تقييم أداء الخوارزميات تم اعتبار بيئة 200m X 200m و 16 حساس ذو موضع ثابت

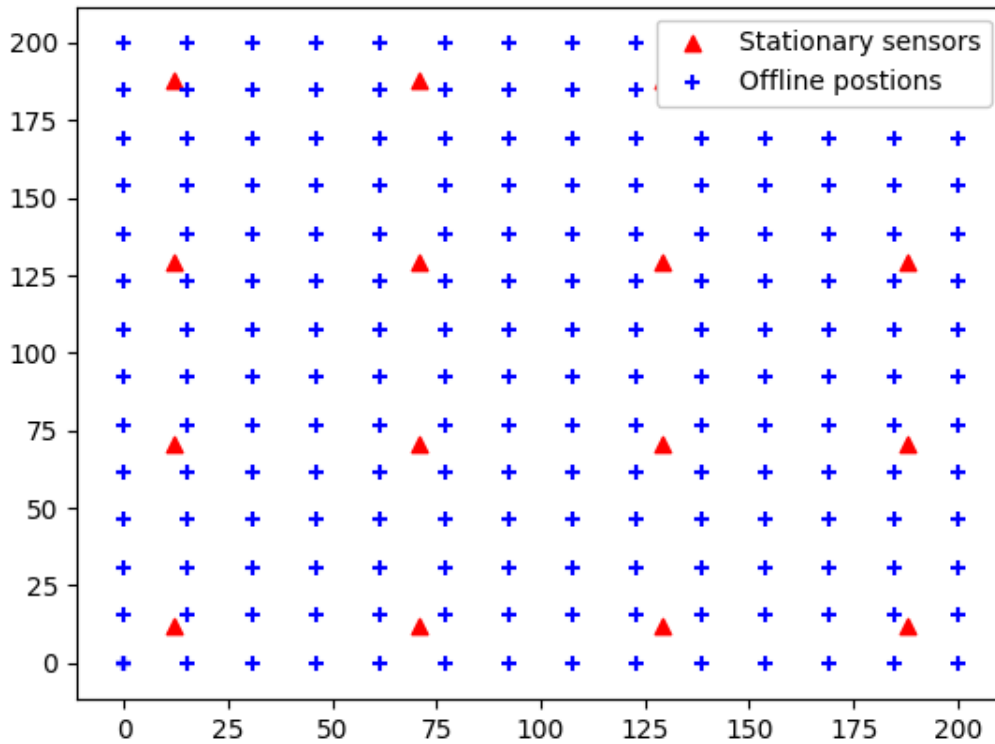
يعد توليد ومحاكاة انتشار الأمواج الراديوية ضمن شبكة الحساسات اللاسلكية أمر مهم جداً لاختبار أداء وتدريب الخوارزميات وغالباً ما تعتمد الأبحاث العلمية هذه الطرق لسهولة إعادة توليدها بظروف عمل مختلفة إذ من المكلف جداً إنشاء معطيات حقيقية لكل حالة اختبار ولكل بيئة ويعد نموذج Okumura من أشهر هذه الطرق لمحاكاة شدة الإشارة ضمن شبكة الحساسات اللاسلكية.

وقد بني النموذج بناء على معطيات عن انتشار الأمواج الراديوية في مدينة طوكيو، يابان ويعطي شدة الإشارة بالعلاقة:

$$\rho_{i,pl} = \rho_0 - 10 n p \log_{10} norm(s_i - p_l) + \varepsilon_{i,l}$$

بوجود هذا النموذج أصبح بالإمكان توليد RSSI (شدة إشارة الراديوية) لكل موقع في الشبكة

وسنولد 100 عينة متوزعة بشكل متوازن في الشبكة للتدريب (offline position) كما يوضح الشكل 2-3 حيث أن هذا العدد يلعب دوراً مهماً في أداء الخوارزميات إذ في غالب الأحيان ما تتحسن الدقة بتزويد عينات التدريب ولكن تم الاكتفاء بهذا العدد نظراً للجهد الكبير لتحصيل مثل هذه البيانات على الواقع، فإنه من غير المنطقي تقييم أداء خوارزمية على عدد كبير في المحاكاة، وفي أرض الواقع لا نستطيع توليد ذلك العدد وقد اعتمدت الدراسة (Sandy Mahfouz 2014) العدد 100 على أنه كافٍ.



الشكل 2—3 توزيع نقاط التدريب ضمن البيئة وأماكن الحساسات

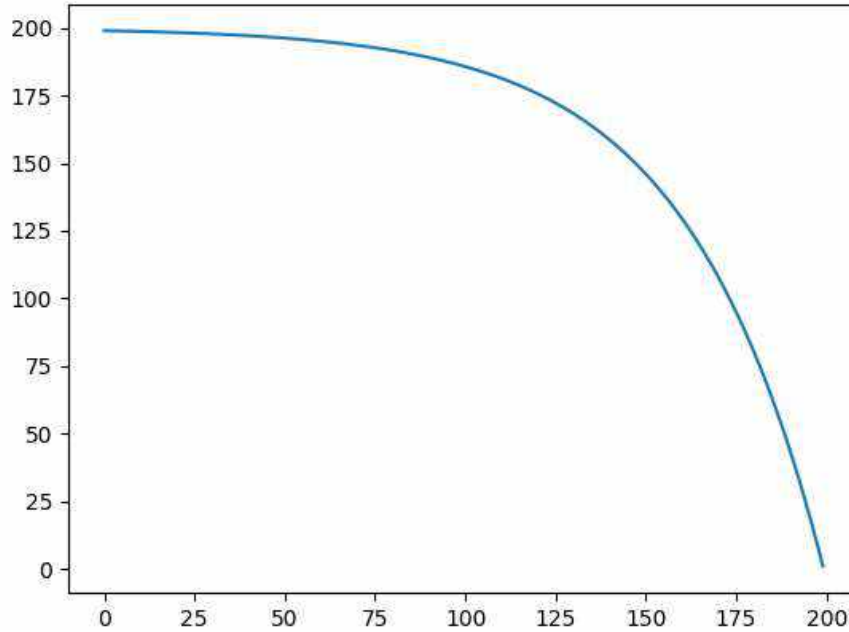
### 3.3.3 إنشاء المسارات الافتراضية

لاختبار أداء الخوارزميات بعد التدريب سنستخدم ثلاثة مسارات مشابهة للدراسة حيث أن المسار الأول هو نموذج الحركة ذو سرعة ثابتة والنموذج الثاني يحاكي تغيرات السرعة والمسار الثالث هو مسار ذو تغيرات كبيرة في التسارع (Sandy Mahfouz 2014)

### 3.3.4 تحديد الموقع وملاحقته

بفرض وجود هدف يتحرك ضمن مسار ما وليكن المسار كما في الشكل 3-3 ونريد التنبؤ بموضع الهدف، ولنفترض أن لدينا نظام تحديد موجود أساساً لتعقب هذا الهدف وعند كل فاصل زمني نستدعي نظام تحديد الموقع وعندها النظام يعطينا الخرج كما في الشكل 3-4. نلاحظ أن نقاط تنبؤ نظام تحديد الموقع المتتالية لا تشكل أي مسار ولديها ترتيب عن القيمة الحقيقية.

في هذه الحالة نلجأ إلى ترشيح أخطاء القياس واستنتاج مسار متصل عن طريق مرشح كالمان



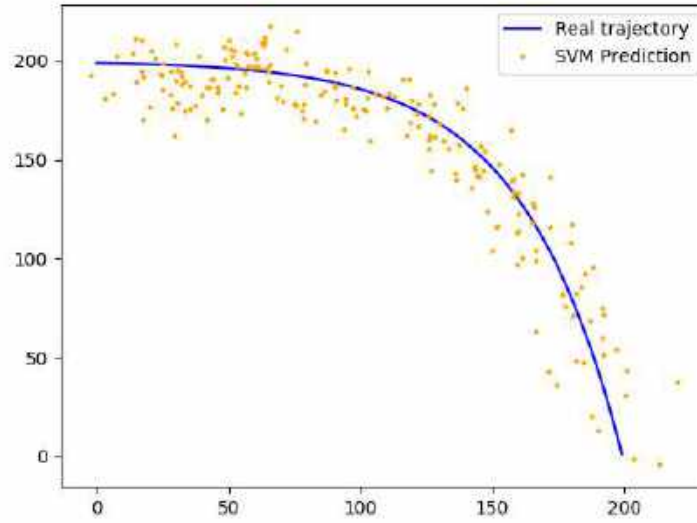
الشكل 3-3 المسار الحقيقي المفروض ضمن البيئة

الذي يعطينا التنبؤ الأمثلي لمسار الجسم المراد تعقبه لنحصل على ملاحقة كما في الشكل 3-5.

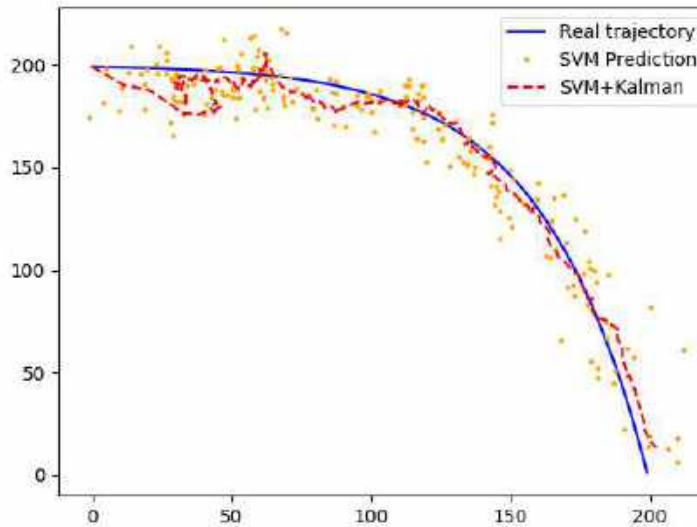
من المهم تزويد مرشح كالمان بمصفوفة تباين أخطاء القياس ونموذج الحركة للجسم المراد تعقبه كمدخلات له، إضافة إلى تحديد ضجيج العملية وهنالك بعض البارامترات التي يمكن تجاهلها كالموضع الابتدائي وإشارة التحكم إذ أن عدم إعطاء الموضع الابتدائي لا يؤثر على



مرشح كالمان وإنما يؤخر تنبؤه المثالي عدداً ما من الفواصل الزمنية.



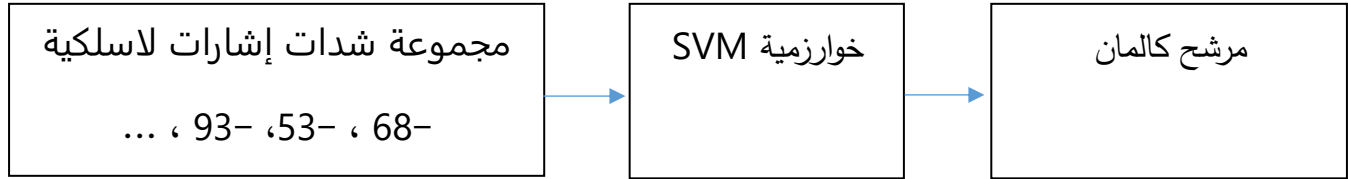
الشكل 3—4 توقعات المسار بحسب خوارزمية تحديد الموقع



الشكل 3—5 ترشيح خرج خوارزمية تحديد الموقع

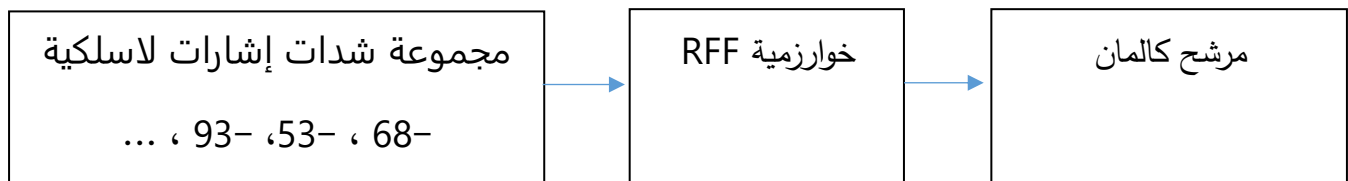
### 3.3.5 نتائج الدراسة Kernel approximation and Kalman Filtering

هذه الدراسة هي تطوير لدراسة سابقة (Saleh 2017) حيث عمل خوارزميات تعلم الآلة يقتصر على تقدير الموقع بناءً على مجموعة شذات إشارة مستقبلية ومن ثم الموقع المتوقع من خوارزمية تعلم الآلة يكون دخلاً لمرشح كالمان لملاحقه جسم متحرك والحصول على تنبؤ أمثلي والدراسات المبنية سابقا كانت تعتمد الشكل التالي:



الشكل 3 — 6 المخطط الصندوقي لتحديد الموقع عن طريق SVM

ولدينا مرشح كالمان يفترض أن خوارزمية التنبؤ هي خوارزمية خطية و SVM تعتمد على kernel لاخطية لذلك حاولنا استخدام طرق تقريب kernel ل SVM ومن أشهر هذه الطرق هي سمات فورييه العشوائية Random Fourier Features كما في الشكل التالي:



الشكل 3 — 7 المخطط الصندوقي لتحديد الموقع عن طريق RFF

وكنتائج نهائية تكون نتائج خرج SVM أفضل من RFF ولكن بعد ترشيح كالمان تكون النتائج الكلية لاستخدام RFF أفضل ويوضح الجدول 1-3 و 2-3 النتائج المستحصلة من SVM ومن ثم RFF بعد ترشيح كالمان:

**الجدول 1—3 الخطأ الوسطي المطلق لخوارزمية SVM**

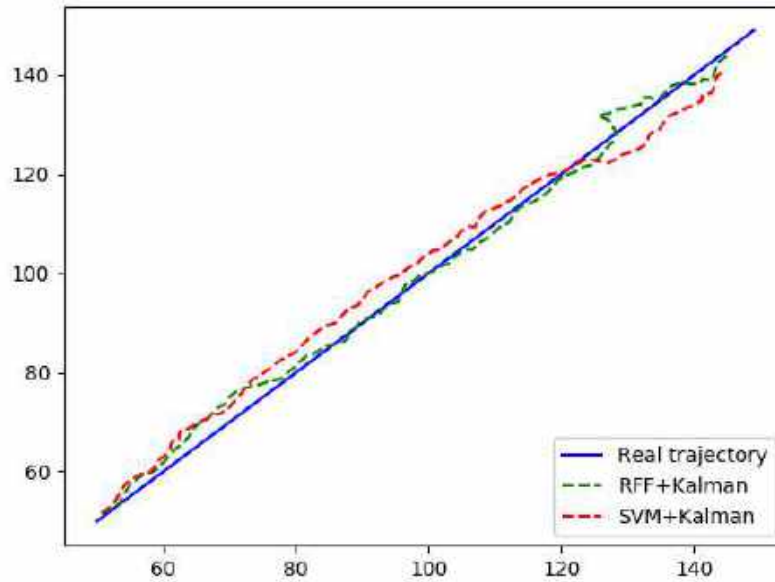
المسار الثالث	المسار الثاني	المسار الأول	
6.76	5.8	4.292	الدرجة الأولى نموذج
6.36	5.5	5.9	الدرجة الثانية نموذج

**الجدول 2 — 3 الخطأ الوسطي المطلق لخوارزمية RFF**

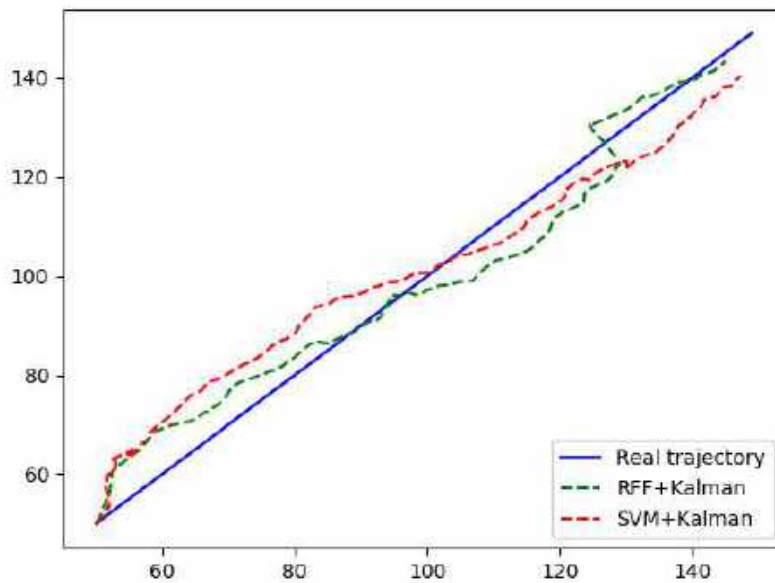
المسار الثالث	المسار الثاني	المسار الأول	
5.05	4.8	1.79	الدرجة الأولى نموذج
4.76	4.67	3.1	الدرجة الثانية نموذج

ولمقارنة النتائج بشكل مرئي تم اعتماد وتوليد مسارات ثلاث بطريقة مشابهة لدراسة (Sandy Mahfouz 2014)

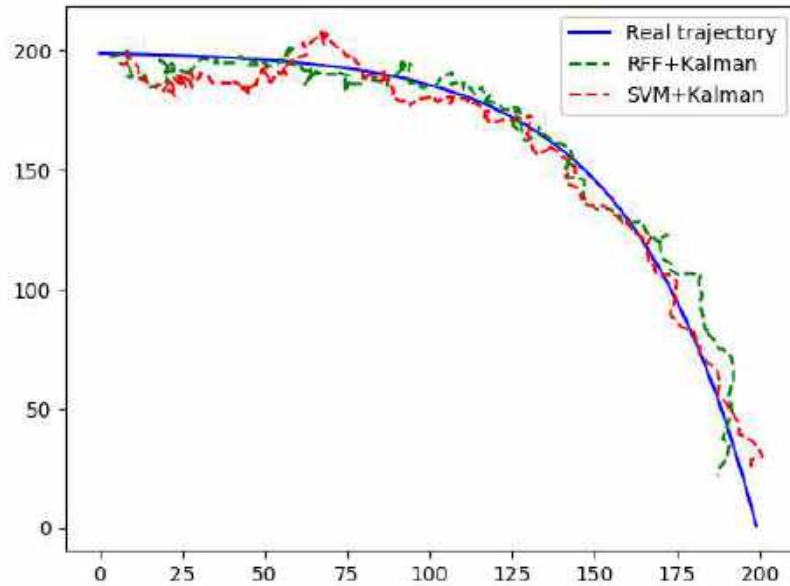
كما تبين الأشكال التالية :



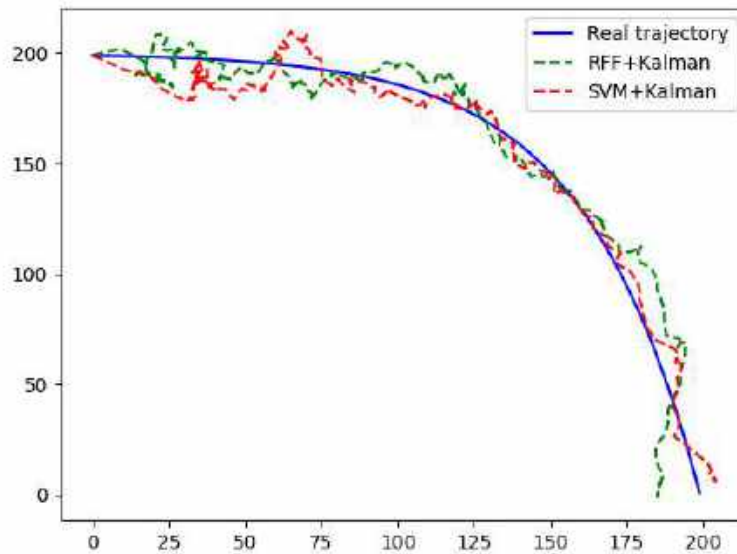
الشكل 3—8 المسار الأول الحقيقي والمتوقع من SVM و RFF بعد استخدام مرشح كالمان من الدرجة الأولى



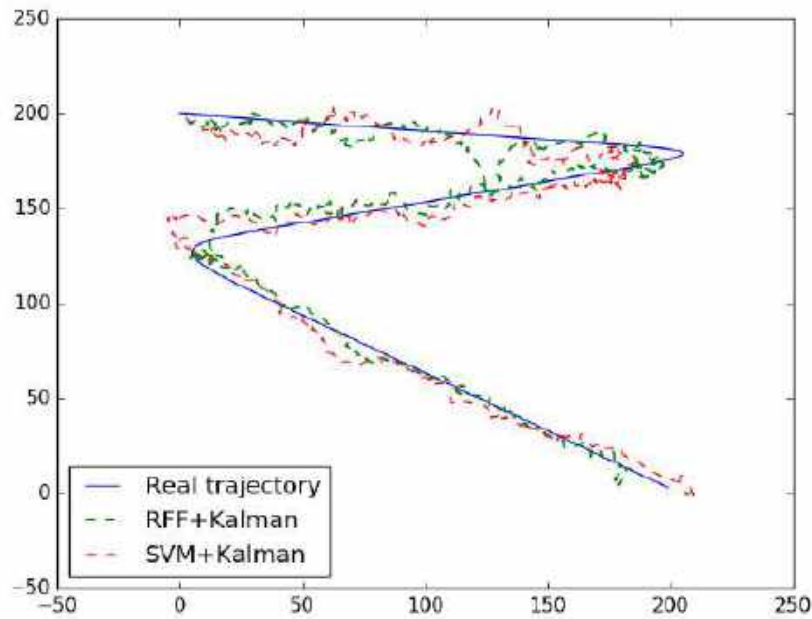
الشكل 3—9 المسار الأول الحقيقي والمتوقع من SVM و RFF بعد استخدام مرشح كالمان من الدرجة الثانية



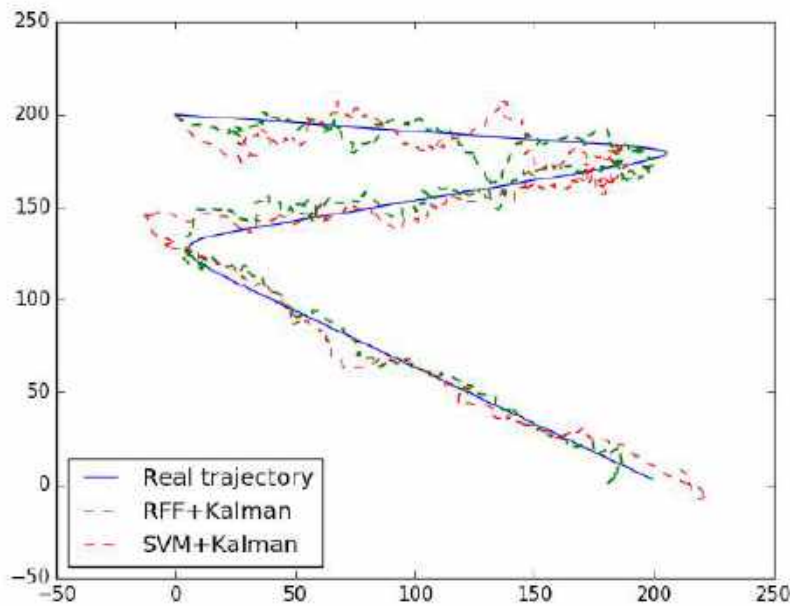
الشكل 3—10 المسار الثاني الحقيقي والمتوقع من SVM و RFF بعد استخدام مرشح كالمان من الدرجة الأولى



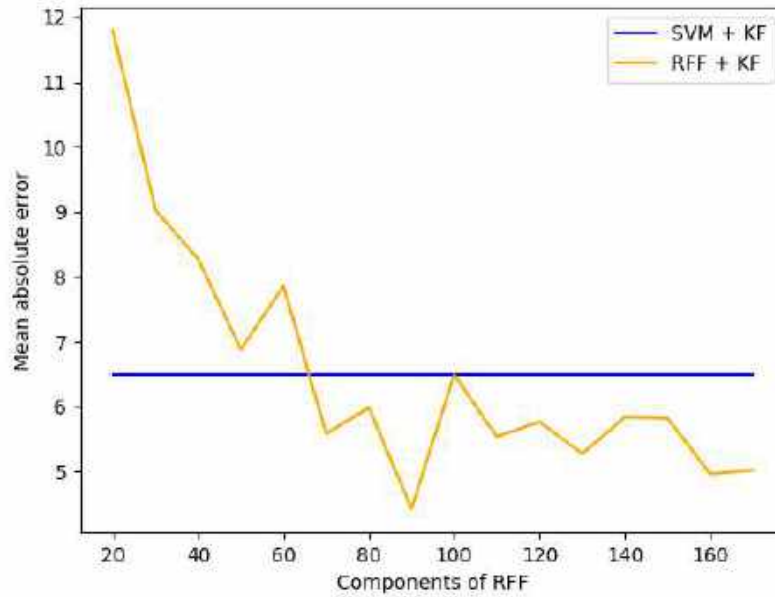
الشكل 3—11 المسار الثاني الحقيقي والمتوقع من SVM و RFF بعد استخدام مرشح كالمان من الدرجة الثانية



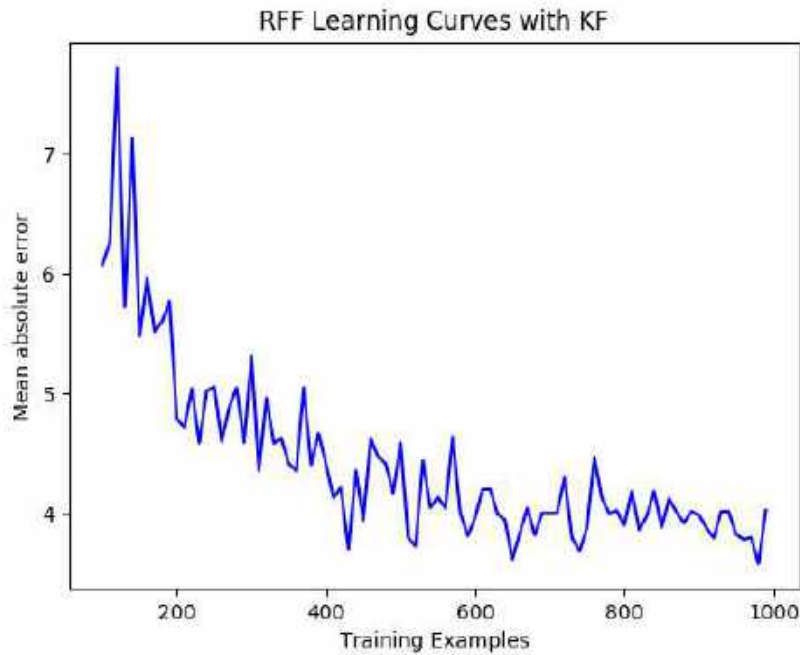
الشكل 3—12 المسار الثالث الحقيقي والمتوقع من SVM و RFF بعد استخدام مرشح كالمان من الدرجة الأولى



الشكل 3—13 المسار الثالث الحقيقي والمتوقع من SVM و RFF بعد استخدام مرشح كالمان من الدرجة الثانية



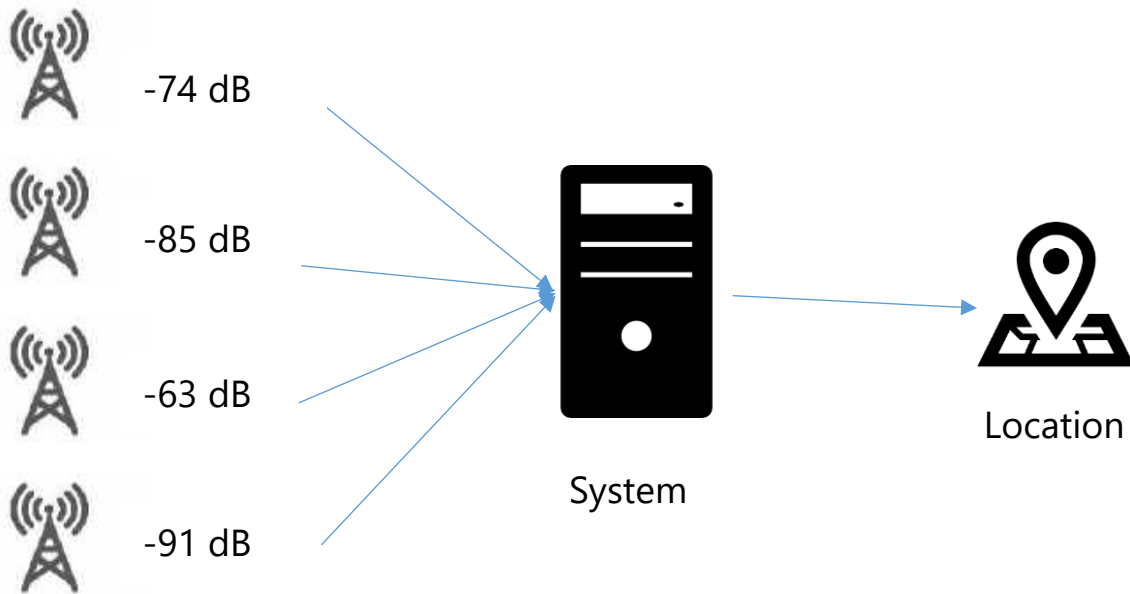
الشكل 3 — 14 الخطا الوسطي المطلق بدلالة عدد عناصر RFF



الشكل 3 — 15 منحنيات التعلم للخطأ الوسطي المطلق لخوارزمية RFF و Kalman Filter

### 3.4 تطبيق عملي على معطيات مزود خدمة اتصالات

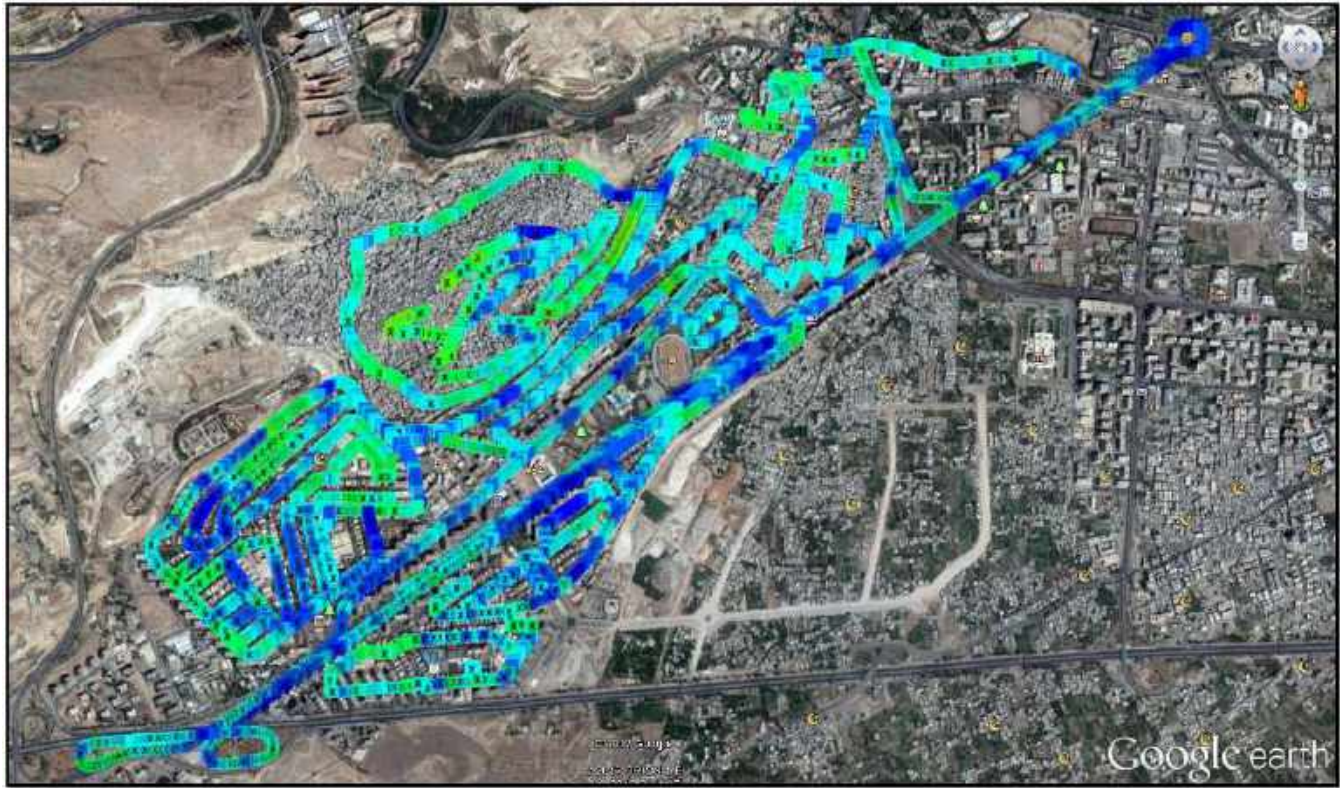
يمكن أن تطبق مفاهيم الدراسة السابقة باستخدام عتاد مناسب وضمن الأبعاد المفروضة، وفي حال كانت الحساسات من نوعية جيدة سنحصل على نتائج المحاكاة مع ترتيبات صغير، وأيضاً يمكن أن نعمم مبدأ شبكة الحساسات على شبكة الاتصالات الخلوية حيث كل محطة قاعدية تبتّ إشارات راديوية في موقع ثابت والمستخدم يستقبل عدة إشارات في نفس اللحظة تصل في الحالات الجيدة إلى سبع محطات خلوية.



الشكل 3 — 16 مبدأ عمل نظام تحديد الموقع

ضمن العمل على المشروع تم التواصل مع شركة مزودة لخدمات الاتصالات سيريتل وشرح فكرة المشروع وعلى أساسها حصلنا على معطيات ضمن منطقة المزة في دمشق على مساحة حوالي  $3.3 \text{ km}^2$  ويبين الشكل 3-17 أماكن توضع هذه النقاط حيث تتكون المعطيات من إحداثيات جغرافية ومن ثم معلومات عن الأبراج الخلوية المرتبطة بإحداثياتها ما وقد تم استحصال حوالي 16 ألف عينة كل عينة كما ذكرنا سابقاً هي عبارة عن إحداثيات جغرافية لموقع ومعلومات شبكة الخلوية المتوافقة معه.



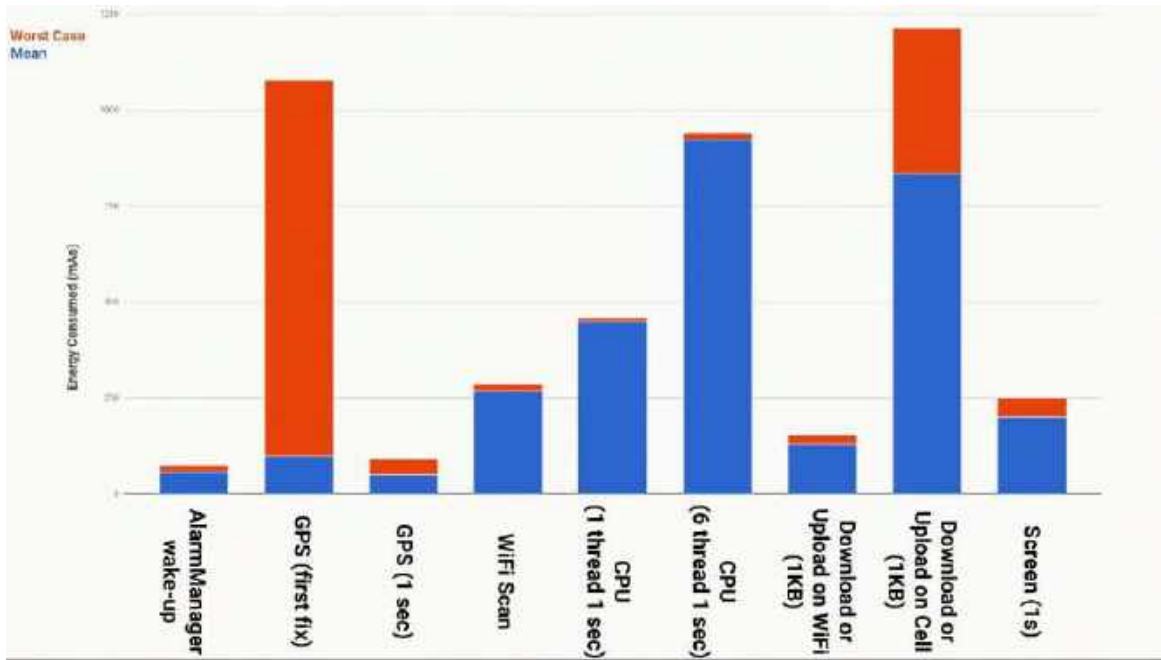


الشكل 3—17 أماكن استحصال العينات

### 3.4.1 أهمية أنظمة تحديد الموقع البديلة

وتكمن أهمية هذه الأنظمة في تكوين حل بديل ل GPS إذ يعد حل موفر للطاقة في حال عدم تطلب التطبيق دقة عالية إذ ذكرت Google في مؤتمرها السنوي I/O 2017 كمية الطاقة التي يستهلكها GPS مقارنة بالفعاليات الأخرى في الشكل 3-18. يزودنا هذا النظام أيضاً بإمكانية تحليل بيانات المستخدمين في أي موضع حيث تحصيل هذه المعطيات لا يعتبر صعب أو مكلف من ناحية استهلاك الطاقة وبالتالي يمكن تحليل مسارات المستخدمين واستخدامها في تطبيقات عديدة

استخدام هذه التقنيات في حالات الخطر وطوارئ إذ يمكن معرفة وجود مستخدمين في مناطق خطر عن طريق شبكة الأبراج الخلوية بالإضافة لتحديد الموقع في المواضع التي لا يصلها إشارات GPS مثل الأنفاق والمنشأة.



الشكل 3 — 18 بعض مصادر استهلاك الطاقة في أجهزة Android من مؤتمر Google I/O 2017

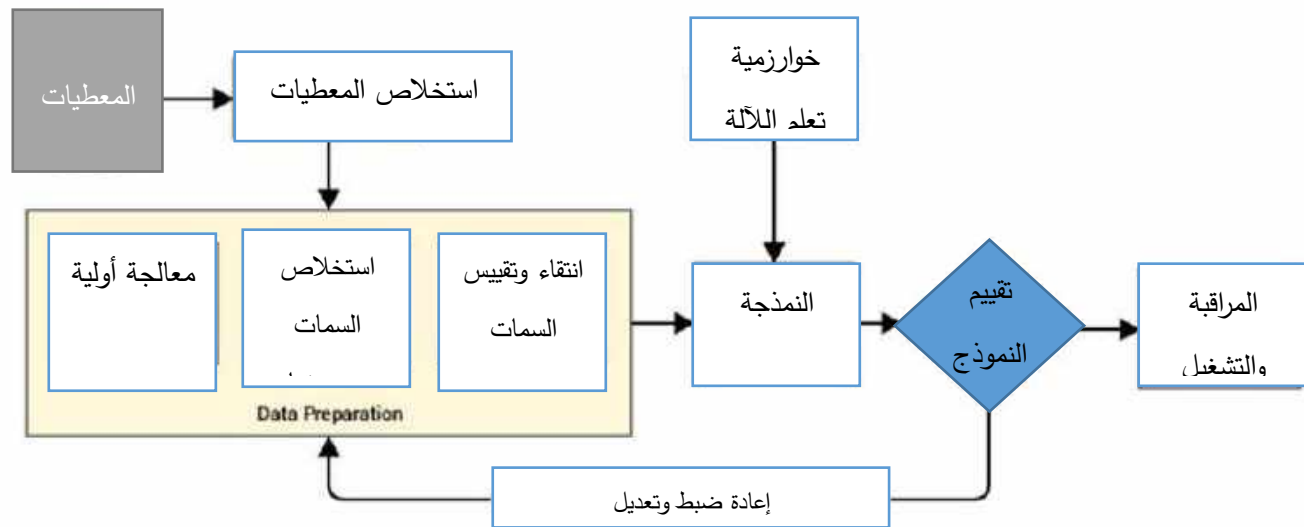
### 3.4.2 بناء نظام تحديد الموقع من معطيات شركة اتصالات

بناء نظام تعلم آلة يحدد الموقع بالاعتماد على معطيات شركة الاتصالات الخلوية يكون بتدريب خوارزميات تعلم الآلة على المعطيات ولكن من غير الممكن إدخال هذه المعطيات كما هي إلى أحد خوارزميات تعلم الآلة حيث أن هذه الخوارزميات تحتاج شكلاً محدداً من السمات المستخلصة. المعطيات من شكل إحداثيات جغرافية بالدرجات ومن ثم معلومات عن الأبراج الخلوية ويجب التحويل للإحداثيات إلى المتر بدلاً من الدرجات لحساب الخطأ بالإضافة إلى أن المعطيات يمكن أن تحوي بعض القيم المفقودة أو غير المرغوب بها وهناك العديد من الأسباب أيضاً جعلت المختصين، في مجال تعلم الآلة وتنقيب المعطيات استنتاج معيار لبناء نظم تعلم آلة بالاعتماد على معطيات واقعية.

#### 3.4.2.1 أنبوب خوارزميات تعلم الآلة Machine Learning Pipeline

أفضل الطرق لحل مشاكل تعلم الآلة أو المسائل التحليلية الواقعية هي استخدام أنبوب تدفق تعلم الآلة بدءاً من أخذ المعطيات إلى تحويلها إلى معطيات ذات دلالات قيّمة باستخدام تقنيات تعلم الآلة (Dipanjan Sarkar 2018)

يتكون أنبوب تعلم الآلة بشكل رئيسي من عناصر متعلقة باسترجاع واستخلاص وتهيئة المعطيات والنمذجة والتقييم ووضع النموذج ضمن بيئة العمل. الشكل 3-19 يوضح نظرة عامة لأنبوب تعلم الآلة المعياري مع وجود الخطوات الأساسية



الشكل 3 — 19 المخطط التدفقي لأنبوب تعلم الآلة

### 3.4.2.1.1 فعاليات أنبوب تعلم الآلة

- **استخلاص المعطيات** وهي بشكل رئيسي تجميع المعطيات واستخلاصها واستحصالها من عدة مصادر ومخازن، وفي مشروعنا تم ذلك من مصدر معطيات واحد.
- **تجهيز المعطيات** في هذه المرحلة نعالج المعطيات بشكل مبدئي وننظفها ونغير فيها كما يجب وأيضا نحلل تلك المعطيات بشكل مبدئي،
- **استخلاص وهندسة السمات** هنا نستخلص السمات المهمة من المعطيات الخام وحتى أنه من الممكن تشكيل سمات جديدة من السمات الموجودة أساسا
- **تقييم وانتقاء السمات** غالباً ما تحتاج السمات إلى إعادة تقييم لمنع خوارزمية تعلم الآلة من الانحياز لبعض السمات وأحياناً نحتاج إلى انتقاء السمات الأفضل، وليست كل سمات هذه المرحلة تم تجربتها، ولكنها لم تحسّن في أداء النموذج في مشروعنا لذلك تم تخطي هذه المرحلة
- **النمذجة** في هذه المرحلة نحن عادة نعطي خوارزمية تعلم الآلة سمات المعطيات وندريب النموذج عموماً لتخفيض دالة الكلفة وبشكل عام الهدف هو تخفيض خطأ التعميم
- **تقييم أداء النموذج** وإعادة ضبط بارامتراته حيث يتم اختبار النموذج على معطيات جديدة لم يتدرب عليها واخذ قياسات الخطأ. إن منهجية الخطأ المتبعة في مشروعنا هي الخطأ الوسطي المطلق وتم أخذ 20% من المعطيات الكلية كعينات اختبار وتقييم بشكل عشوائي وليس مرتب من كافة المعطيات 16 ألف عينة وضبط البارامترات لإعطاء أفضل النتائج.
- **التشغيل والمراقبة** إذ أنه بعد بناء النموذج يتم تصديره بشكل مناسب ليتم استخدامه ضمن بيئة العمل وأيضا مراقبة أدائه

#### 3.4.2.1.2 النتيجة

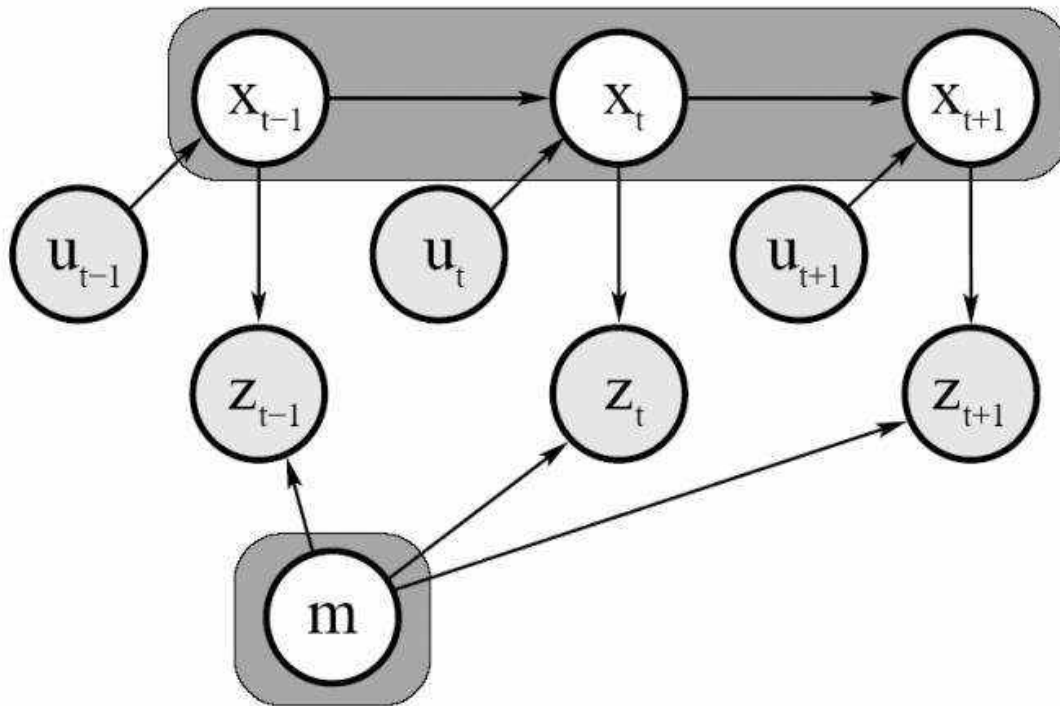
أداء نظام الآلة المبني لمشكلة التموضع استطاع أن يحدد المكان بخطأ وسطي يتراوح بين 10 متر و14 متر وهذا رقم يعد جيد جداً في شبكات GSM ونستطيع تخفيض الخطأ أكثر ولكن نحتاج إلى قدرات حوسبة أعلى.

## الفصل الرابع

### 4 خوارزمية SLAM

## 4.1 مقدمة

أحد أكثر الأمور أهمية في علم الروبوتيك هي بناء الخرائط لأنها تكون اللبنة الأساسية التي تعتمد عليها الخوارزميات الأخرى مثل التخطيط والتموضع و... تعرف خوارزمية بناء الخرائط بـ SLAM اختصار لـ Simultaneous Localization and Mapping حيث تبرز أهميتها عندما يدخل الروبوت في بيئة لا يعلم عنها شيء وإنما لديه قياسات للبيئة المحيطة وأوامر التحكم، اسم الخوارزمية يصف المسألة حيث الروبوت عند استخدام SLAM يحصل خريطة حول البيئة بالتزامن أيضا مع تقدير لموقعه ضمن تلك الخريطة، لذلك تعد من المسائل الصعبة في هذا المجال (Søren Riisgaard 2005)



الشكل 4 — 1 عمل خوارزمية SLAM

الشكل 4-1 بين عملية SLAM ضمن عدة خطوات زمنية وحيث  $x$  تمثل الموقع و  $u$  أوامر التحكم و  $z$  القياسات للبيئة المحيطة و  $m$  الخريطة المبنية



سنقدم في هذا الفصل مبدأ SLAM باستخدام مرشح كالمان اللاخطي Extended Kalman Filter الذي يفيد في تقدير الموقع مع ارتياب ما بواسطة تمثيل الهدف بفضاء حالة. مع العلم أن SLAM ممكن أن تطبق بطرق عدة ويوجد هنالك الكثير من أنواع العتاد الذي يمكن استخدامه.

## 4.2 لمحة تاريخية

طورت slam بشكل أساسي من قبل Hugh Durrant-Whyte و John J. Leonard بالاعتماد على عمل سابق من قبل Smith و Self و Cheeseman.

خوارزمية SLAM تحتوي على عديد من الخطوات:

تحديد نقاط العلام، تجميع المعطيات، تقدير الموضع، تحديث الموضع ونقاط العلام. ويوجد أكثر من طريقة لعمل كل خطوة من الخطوات.

## 4.3 العتاد

تستخدم SLAM بشكل أساسي المعلومات التي تفيد في كمية المسافة التي اجتازها الروبوت ودورانه Odometry التي من ممكن استحصالها بشكل صريح عن طريق مرمزات عجلات الروبوت أو استخراجها من قيم حساسات أخرى مثل قيم حساس الليزر عن طريق خوارزمية تعرف باسم Scan matcher.

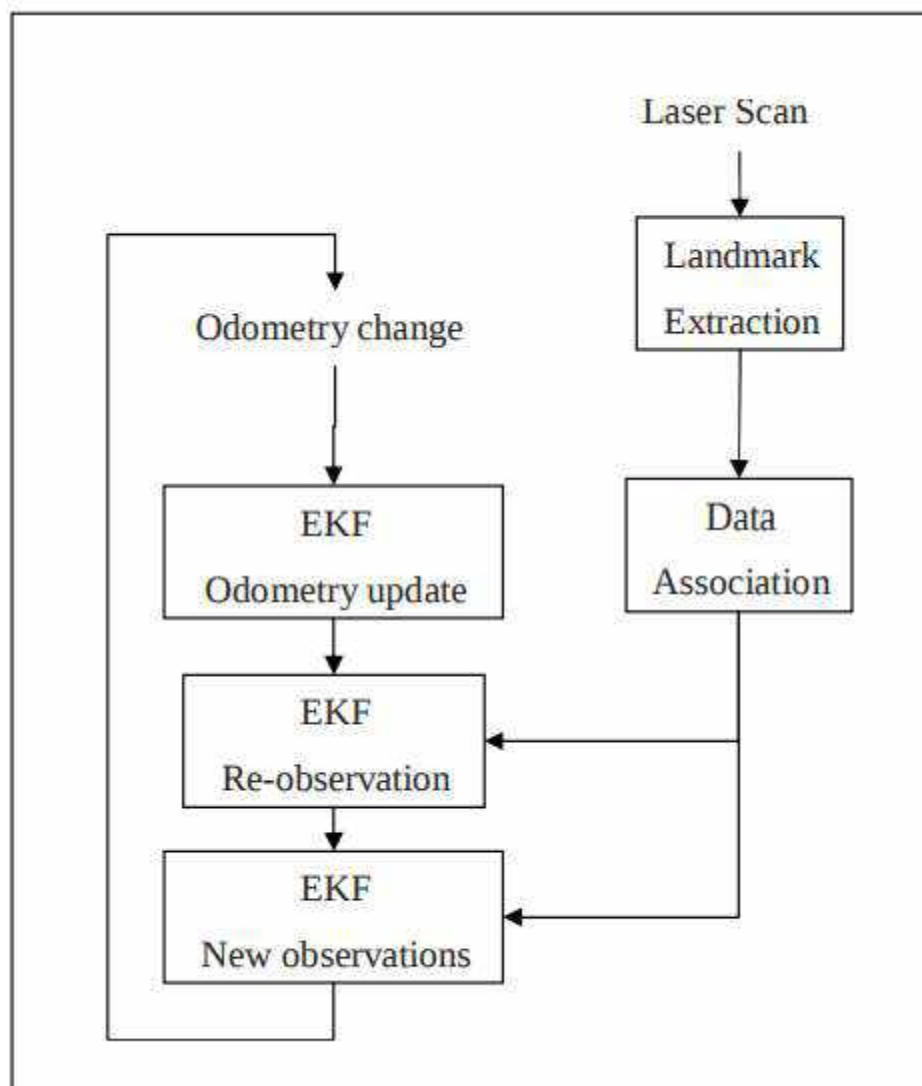
وأيضاً تعتمد على جهاز حساس للمسافة الذي يكون غالباً حساس ليزري وبالتالي خصائص الحساس الليزري المستخدم ونوعيته تلعب دوراً في قيمة ارتياب الخطأ في الخريطة المبنية. مع العلم أنه يوجد خيارات أخرى كالسونار و Stereo-Camera.

## 4.4 خوارزمية عمل SLAM

الهدف هنا هو إيجاد موقع الروبوت، بما أن Odometry للروبوت يكون عادةً غير دقيق لا يمكننا اعتماد فقط عليه لذلك نستخدم المعطيات من حساس الليزر لنستفيد منها في تحديد الموقع



أيضاً. هذا الأمر ممكن عن طريق استخلاص سمات مميزة بالبيئة وتحديد موقع الروبوت نسبةً لها. مرشح كالمان الاخطي Extended Kalman Filter هو المسؤول عن تحديد أين يظن الروبوت نفسه بالاعتماد على تلك السمات واستخلاص تنبؤ أمثلي حيث أن EKF يأخذ بعين الاعتبار الارتياحات الناتجة عن عملية القياس وأماكن وجود السمات. السمات المميزة سنطلق عليها اسم نقاط العلام Landmarks.

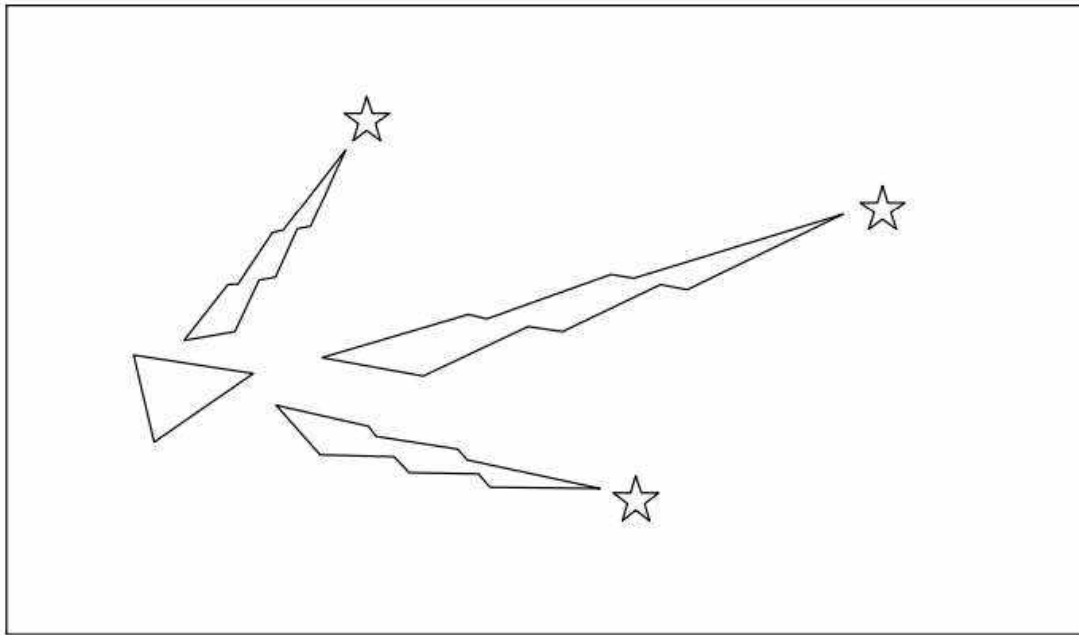


الشكل 4 — المخطط الصندوقي لـ SLAM

الميزة الأساسية بمرشح كالمان هو أنه يأخذ بعين الاعتبار الارتياح في عملية القياس في حساباته ويعطي الموقع الأكثر دقة سواء كان الارتياح ناتج عن Odometry أو ارتياح ناتج عن تحديد موقعه بالنسبة لنقاط العلام.

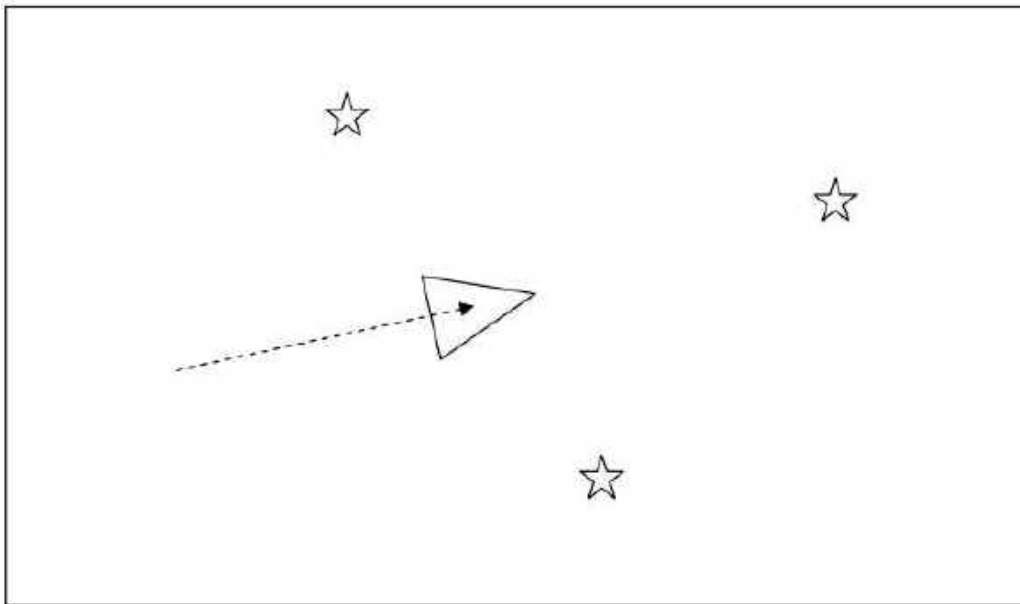
مثال يوضح كيفية استخلاص الموقع من نقاط العلام:

لدينا الروبوت الممثل بالمثلث، النجوم نقاط العلام، الأشعة هي قيم حساس المسافة



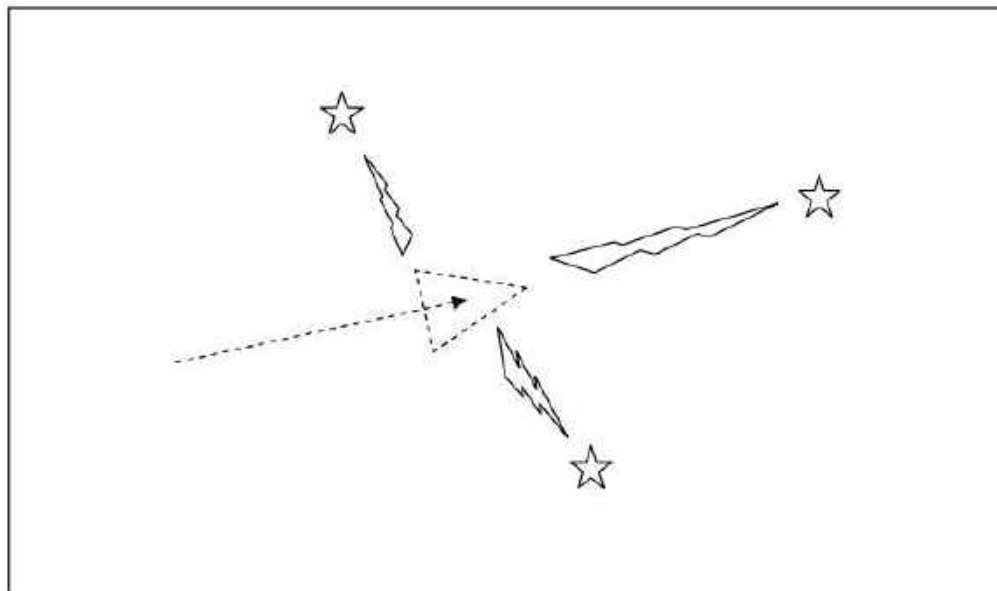
#### الشكل 4 — 3 المبدأ النظري لإشارات حساس المسافة مع نقاط علام

عند تحرك الروبوت يعتقد أنه في مكان ما اعتماداً على Odometry



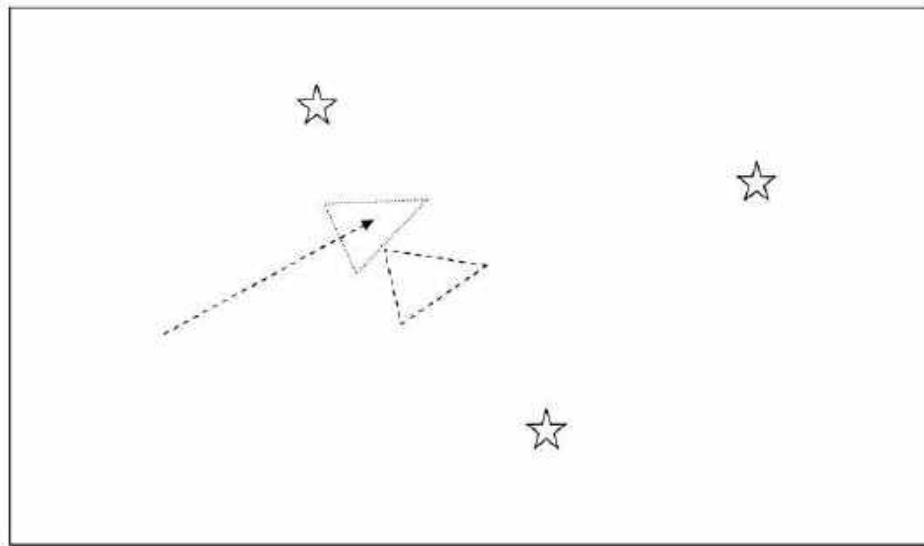
الشكل 4 — 4 الشكل الروبوت ونقاط العلام

عندما يقيس الروبوت مرة أخرى موضعه بالنسبة لنقاط العلام عن طريق حساس المسافة قد يكون الموضع غير مطابق لموقعه المقاس عن طريق Odometry



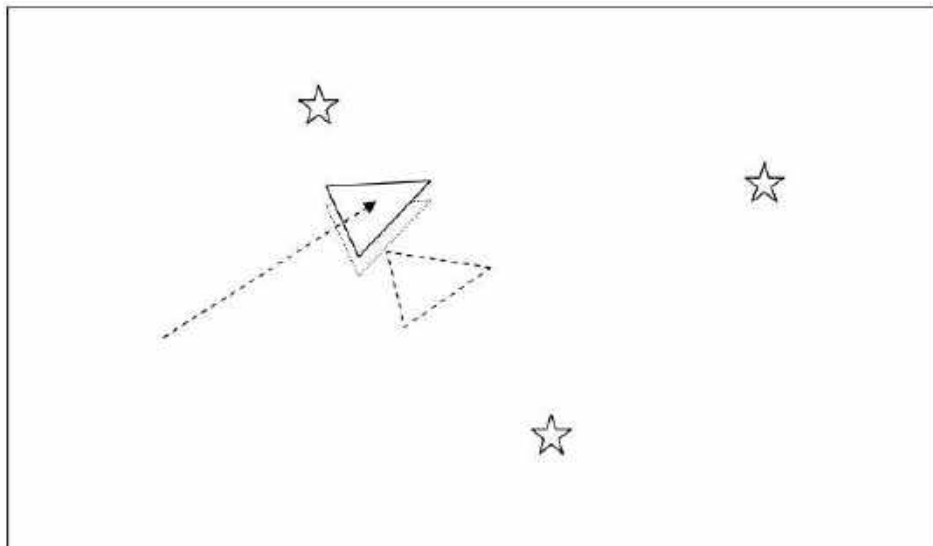
الشكل 4 — 5 إشارات حساس المسافة

بما أن ارتياب حساس المسافة عادة أقل من Odometry يتم تحديث موقعه بناءً على ثقته بالمعطيات الواردة من حساس المسافة ( المثلث المتقطع هو المكان الذي كان يعتقد الروبوت أنه موجود فيه حسب حساس المسافة)



الشكل 4 — 6 توقع بحسب Odom

قد لا نحصل على موقع الروبوت الحقيقي بالنهاية وإنما أفضل تقدير بخطأ مقبول



الشكل 4 — 7 فرق بين الموقع المتوقع والحقيقي

### 4.4.1 نقاط العلام Landmarks

نقاط العلام هي سمات مميزة في البيئة يستخدمها الروبوت لتحديد موقعه. ونوع نقاط العلام يعتمد على البيئة الموجودة فيها الروبوت. نقاط العلام يجب أن نقدر على مشاهدتها أكثر من مرة من عدة أماكن وزوايا، وسهلة التمييز في عدة لحظات زمنية.

وبالتأكيد هي نقاط ثابتة ضمن البيئة لان في حال تحركها يظن الروبوت أن موقعه قد اختلف. يوجد عدة طرق لاستخلاص نقاط العلام تختلف بحسب العتاد المستخدم والبيئة أي قد نعتد على النقاط البارزة في البيئة كما في طريقة Spike أو على الجدران المستوية أحيانا

### 4.4.2 ترابط المعلومات Data Association

المسألة هنا هو مطابقة نقاط العلام المرصودة من عدة قياسات من حساس المسافة وهذا يدعى بإعادة رصد نقاط العلام re-observing landmarks المشاكل الرئيسية ممكن أن تظهر في ربط المعلومات

- يمكن ألا تعيد رصد نقاط العلام في كل خطوة زمنية.
  - يمكن أن ترصد شيء كنقطة علام وتفشل في إعادة رصده مجدداً.
  - يمكن أن تربط نقطة علام بشكل خاطئ مع نقطة علام رصدها مسبقاً.
- المشكلة الأخيرة تعد كارثية بالنسبة ل SLAM حيث أن الروبوت سيعتقد نفسه في مكان مختلف تماماً عن موقعه الحقيقي.

وللتخلص من تلك المشاكل يتم اعتماد سياسة ترابط المعلومات بحيث نفترض أن لدينا قاعدة معطيات نخرن فيها نقاط العلام التي رصدناها. القاعدة الأولى هنا ألا نستخدم نقطة علام موجودة إلا إذا تم رصدها أكثر من عدد  $N$  من المرات بحيث يتم إقصاء نقاط العلام السيئة. يتم ربط نقطة علام مرصودة حالياً مع قاعدة البيانات عن طريق خوارزمية Nearest Neighbor التي تعتمد على المسافة الإقليدية ( يمكن اعتماد طريقة غير المسافة الإقليدية ).

ومن ثم يتم التحقق من صحة نقطة العلام تلك عن طريق EKF حيث أن المرشح لديه قدرة على تمثيل عدم اليقين Uncertainty.

### 4.4.3 تطبيق مرشح كالمان الاخطي في SLAM

مرشح كالمان الاخطي يستخدم ليقدر موقع الروبوت أمثلياً من معطيات Odometry و معطيات نقاط العلام بحيث يستطيع أيضاً أن ينمذج ارتياب كل منهما واستخدامها في معادلاته. الاختلاف هنا في استخدام مرشح كالمان أنه ليس لدينا خريطة ليتم حساب الموقع لها. وهذا يستدعي بعض التغيير في مصفوفات كالمان

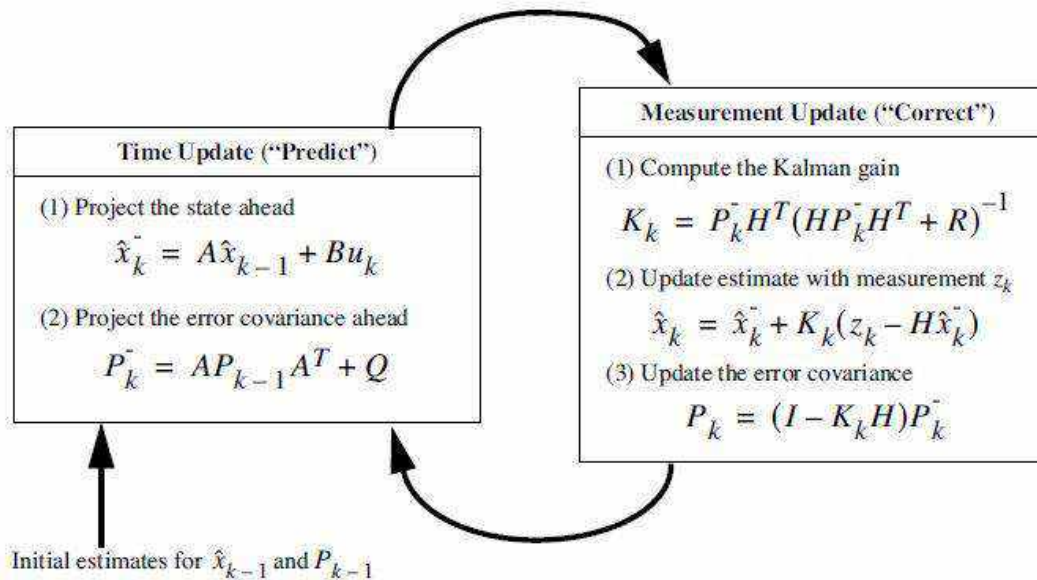
الخطوات العملية :

بعد أن يتم استخراج نقاط العلام وربط المعطيات في مكان ما Slam يقوم بـ:

1. تحديث الموقع بناءً على معطيات Odometry
2. تحديث الموقع بناءً على نقاط العلام المرصودة
3. إضافة نقاط علام جديدة إلى الموقع الحالي

#### 4.4.3.1 تطبيق معادلات كالمان

معادلات كالمان تصنف القسمين إلى جزء معني بعملية تحديث قيم المصفوفات عند الحصول على قياس جديد وقسم معني بالتنبؤ بحالة الروبوت:



الشكل 4 — 8 مراحل مرشح كالمان في SLAM

حيث أن التقدير الابتدائي يأتي من معطيات Odometry

#### 4.4.3.1.1 مصفوفات كالمان

- **مصفوفة الحالة  $X$**

تعد أكثر المصفوفات أهمية إذ تمثل موقع الروبوت ودورانه وإحداثيات نقاط العلام. وتكون مصفوفة عامودية:

$x_r$
$y_r$
$\theta_r$
$x_1$
$y_1$
...
...
$x_n$
$y_n$

- **مصفوفة التغير  $P$**

مصفوفات التغير تعنى بالعلاقات بين المتحولات ومقدار الترابط بينهم وتحتوي على تغير موقع الروبوت وتغير في نقاط العلام

- **رج كالمان  $K$**

وهو مقدار يعبر عن مقدار ثقتنا بقياس نقاط العلام

- **مصفوفة يعقوبي لنظام القياس  $H$**

وهو عبارة عن تحويل من فضاء قياسات حساس المسافة إلى فضاء الموقع وتحتوي معادلات تحويل تختلف حسب العتاد المستخدم

- **مصفوفة اليعقوبي لنظام التنبؤ  $A$**

هي المصفوفة التي تحوي معادلات تحويل لتعطينا حالة الروبوت الراهنة من الحالة السابقة بالإضافة إلى أوامر التحكم.

- **مصفوفات يعقوبي خاصة ب  $Slam$**

يوجد هنالك مصفوفتين خاصتين ب  $Slam$  في مرشح كالمان الاخطي وهما مصفوفة تنبؤ بموقع نقاط العلام من موقع الروبوت ومصفوفة التنبؤ بموقع نقاط العلام من عملية القياس

- **مصفوفة ضجيج الحالة  $W, Q$**   
وهنا يتم تمثيل الضجيج كضجيج غاوسي وأخذه بعين الاعتبار عند الحسابات في معادلات كالمان.
- **مصفوفات ضجيج القياس**  
وهي مصفوفات تحتوي على قيم الارتيايات الحاصلة في عملية القياس أي الثقة في عملية القياس وتختلف حسب دقة كل عتاد ويعد من الصعب تقديرها بدقة.



## الفصل الخامس

### 5 نظام وبيئة Android

## 5.1 مقدمة

Android هو أساسا نظام تشغيل مفتوح المصدر لأجهزة الهواتف النقالة، مبني على نواة لينوكس، يدعم تعدد التطبيقات. أي من الممكن أن يعمل به أكثر من تطبيق في نفس الوقت وتعمل كل التطبيقات فيه بشكل متساوي من ناحية مشاركة موارد الجهاز. وهو مجاني بحيث أن أي شخص يمكن أن يقوم بالبرمجة فيه والتعديل عليه، كما أن لغة جافا تستعمل واجهات مرئية XML سهلة الاستخدام لذا ستجعلنا نقوم بعمل برامج Android بسهولة (Google 2017)



الشكل 5 — 1 شعار Android

## 5.2 تاريخ نظام Android

الأعضاء الرئيسيين المشاركين بتأسيس نظام Android هم Andy Rubin و Nick Sear و Chris White

كانت الأهداف الأولية لشركة تطوير نظام معالجة متقدم للكاميرات الرقمية وعندما أدركت الشركة أن سوق أجهزة الموبايل ذو طلب متزايد توجهت لإنتاج نظام معالجة هواتف ذكية ينافس نظام Microsoft و Symbian

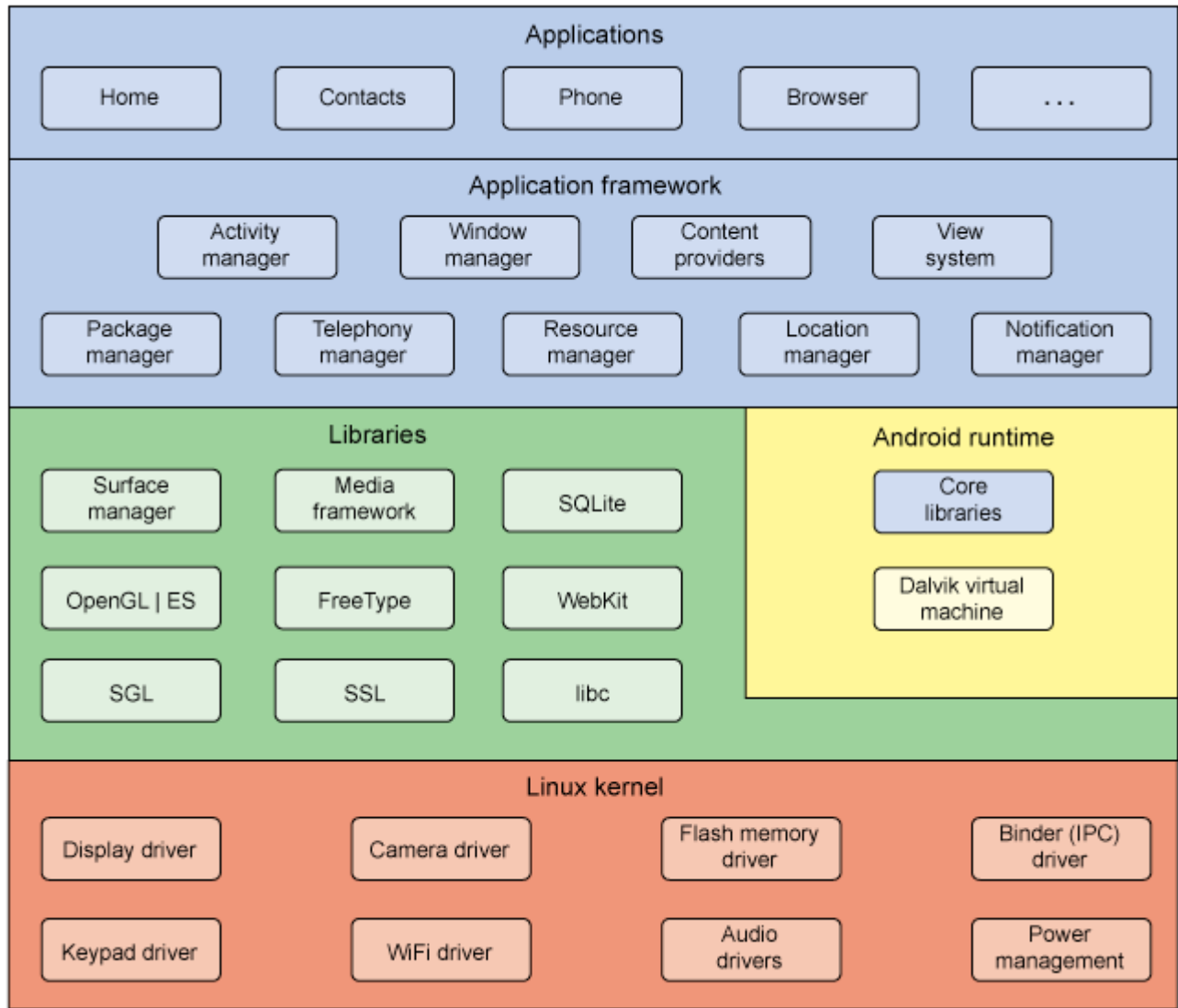
طور فريق من شركة Google تحت إشراف Andy Rubin منصة للجهاز الخليوي مدعومة بنواة Linux وقد سوقت غوغل هذه المنصة على وعد التزود بنظام مرن ومحدث.

واستمرت التكهّنات حول نية شركة Google بالدخول إلى سوق اتصالات الهاتف المحمول حتى عرضت التصميم الأول لجهاز Sooner وهو يشبه هاتف Black Berry ولكن دون شاشة لمس وأعيد هندسته لاحقا ليدعم شاشات اللمس وليتنافس مع الأجهزة الأخرى ك iPhone في الخامس من تشرين الثاني 2007 كشفت Android إنتاجها لأول منصة جهاز موبايل مبني على نواة Linux بالإصدار 2.6.25. أطلق في 22 تشرين الأول 2008 الموبايل التجاري الأول الذي استخدم نظام Android واسمه HTC Dream

### 5.3 مميزات Android

تعتبر الأجهزة ذات نظام تشغيل Android أجهزة متكاملة حيث لم تعد مقتصرة على إجراء مكالمات هاتفية فقط بل أصبحت تدعم آخر صيحات التكنولوجيا بالإضافة إلى العديد من الميزات وهي:

- **التخزين:** يستخدم Android لتخزين البيانات SQLite أو نسخ خفيفة من قواعد البيانات المترابطة.
- **نظم الاتصال:** يدعم Android كافة أنظمة الاتصالات: GSM 4G LTE Bluetooth Wi-Fi ...
- **الرسائل:** يدعم خدمة الرسائل القصيرة، SMS خدمة رسائل الوسائط المتعددة MMS
- **دعم اللغات:** يدعم Android العديد من اللغات البشرية.
- **متصفح الويب:** المتصفح الموجود على Android مبني على واجهة Web Kit مقترن مع محرك الجافا سكربت في متصفح كروم.
- **دعم الوسائط:** يدعم النظام كل الصيغ القياسية للوسائط Media من صور وأصوات وفيديو.
- **المجانية:** يعتبر نظام Android أكثر نظام مجاني حيث أن عدد التطبيقات المجانية فيه يفوق عدد التطبيقات المدفوعة.
- **أدوات الواجهة:** Widgets هي تطبيقات صغيرة توجد على سطح المكتب مثل: تطبيق القدر المتبقي من طاقة البطارية، التقويم... الخ.



الشكل 5 — 2 المكونات الأساسية لنظام تشغيل Android

## 5.4 بيئة تطوير Android Studio



### الشكل 5 — 3 شعار Android Studio

وهو بيئة التطوير الرسمية IDE Integrated Development Environment لأجهزة Android تم الإعلان عنه في 2013/5/16 في مؤتمر Google I/O، وهذه البيئة هي مجانية تماما. Android Studio كان في مراحل الاختبار الأولى بدءا من الإصدار 0.1 في 5/2013، بعدها دخل في مرحلة الإصدار التجريبي beta stage بدءا من الإصدار 0.8 التي تم إصدارها في 6/2014. أول نسخة مستقرة من هذا البرنامج أصدرت في 12/2014، والتي كان رقمها Android 1.0. Studio صمم خصيصا لتطوير برنامج، Android وهو متاح للتنزيل مجانا لجميع أنظمة Eclipse Android Development ، ويمكن استخدام Linux، MacOS، Windows التشغيل (ADT) Tools كبديل عنه.

### 5.4.1 بعض مزايا Android Studio

- عادة الاستخدام والإصلاح السريع للأخطاء.

- وجود محاكي سريع لتجريب التطبيقات سواء كان افتراضيا أو عن باستخدام جهاز حقيقي
- Lint tools والتي تستخدم من أجل مراقبة الأداء، التوافق مع الإصدارات الأخرى، وكشف المشاكل (حيث أن Lint tools هي مجموعة من أدوات نظام Unix تستخدم للإشارة إلى مكان الأخطاء والبرامج المشبوهة).
- دعم إمكانية بناء وتصميم تطبيقات للأجهزة القابلة للارتداء التي تعمل بنظام Android Wear
- Android Things دعم تطوير تطبيقات إنترنت الأشياء على بعض اللوحات التطويرية بلغة Java

## 5.4.2 متطلبات تشغيل Android Studio

### Windows

- Microsoft® Windows® 7/8/10 (32- or 64-bit)
- 3 GB RAM minimum; 8 GB RAM recommended; plus 1 GB for the Android Emulator
- 2 GB of available disk space minimum; 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

### Mac

- Mac® OS X® 10.10 (Yosemite) or higher, up to 10.13 (macOS High Sierra)
- 3 GB RAM minimum; 8 GB RAM recommended; plus 1 GB for the Android Emulator
- 2 GB of available disk space minimum; 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

### Linux

- GNOME or KDE desktop  
Tested on Ubuntu® 14.04 LTS, Trusty Tahr (64-bit distribution capable of running 32-bit applications)
- 64-bit distribution capable of running 32-bit applications
- GNU C Library (glibc) 2.19 or later
- 3 GB RAM minimum; 8 GB RAM recommended; plus 1 GB for the Android Emulator
- 2 GB of available disk space minimum; 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

الشكل 5 — 4 متطلبات Android Studio

## 5.5 بيئة Android

تم تصميم نظام Android بشكل أساسي لتخديم الأجهزة ذات الشاشات اللمسية للأجهزة الذكية والأجهزة اللوحية وهو نظام تشغيل مفتوح المصدر Open source مما يسمح لمصممي الأجهزة بتكيفه ليتلاءم مع منتجاتهم ومما يزيد من قوة هذا النظام هو وجود عدد كبير من التطبيقات المتاحة مجاناً لمستخدمي نظام التشغيل هذا.

تتم كتابة تطبيقات Android باستخدام لغة جافا وتتم ترجمة الشيفرة المصدرية مع البيانات المتعلقة بالبرنامج باستخدام أدوات Android SDK إلى حزمة ذات لاحقة apk

عندما يتم تحميل التطبيق على جهاز Android يتم إنشاء onCreate كل تطبيق وإعطائه صلاحيات أمنية خاصة والذي يدعى Security Sandbox

نظام Android هو نظام Linux متعدد المستخدمين، حيث يكون كل تطبيق من التطبيقات العاملة هو مستخدم مستقل.

يتم إعطاء كل تطبيق من التطبيقات العاملة رقماً فريداً Unique ID يتم التعامل مع هذا الرقم من قبل نظام التشغيل فقط وهو غير معروف بالنسبة للتطبيق، ويقوم نظام التشغيل بإعطاء صلاحيات للملفات التابعة لتطبيق ما، تكون هذه الصلاحيات مرتبطة بالرقم المعرف ID لهذا التطبيق حيث يمكن لهذا التطبيق فقط التعامل مع الملفات.

كل إجراء Process يتم تنفيذه على Virtual machine خاصة وبذلك يتم تنفيذ كل تطبيق بمعزل عن التطبيقات الأخرى ويقوم نظام التشغيل باستبدال تطبيق ما عندما يطلب أي من مكوناته التنفيذ execution

في هذه الطريقة يقوم نظام التشغيل بإعطاء أقل صلاحية ممكنة للتطبيقات، بحيث يمكن للتطبيقات الولوج فقط إلى المكونات التي يحتاجها لإتمام عمله ليس أكثر، وهذا يساعد في خلق بيئة عمل آمنة لا يمكن للتطبيق فيها الولوج إلى مصادر لم يتم إعطاؤه صلاحيات عليها.

يمكن لتطبيقات أن يتشاركان في المصادر والبيانات نفسها من خلال إعطاء هذين التطبيقين نفس الرقم المعرف ID أو من خلال طلب أحد التطبيقات صلاحية الدخول إلى مصادر وبيانات تطبيق آخر (مثل الرسائل، أرقام الهاتف..).

## 5.5.1 مكونات التطبيق

وهي كتل البناء الأساسية لتطبيق Android كل منها يشكل نقطة دخول مختلفة لنظام التشغيل إلى التطبيق أو يلعب دوراً معيناً وخاصة في تعريف مجمل التصرفات المتعلقة بالتطبيق، هناك أربع أنواع من المكونات وهي:

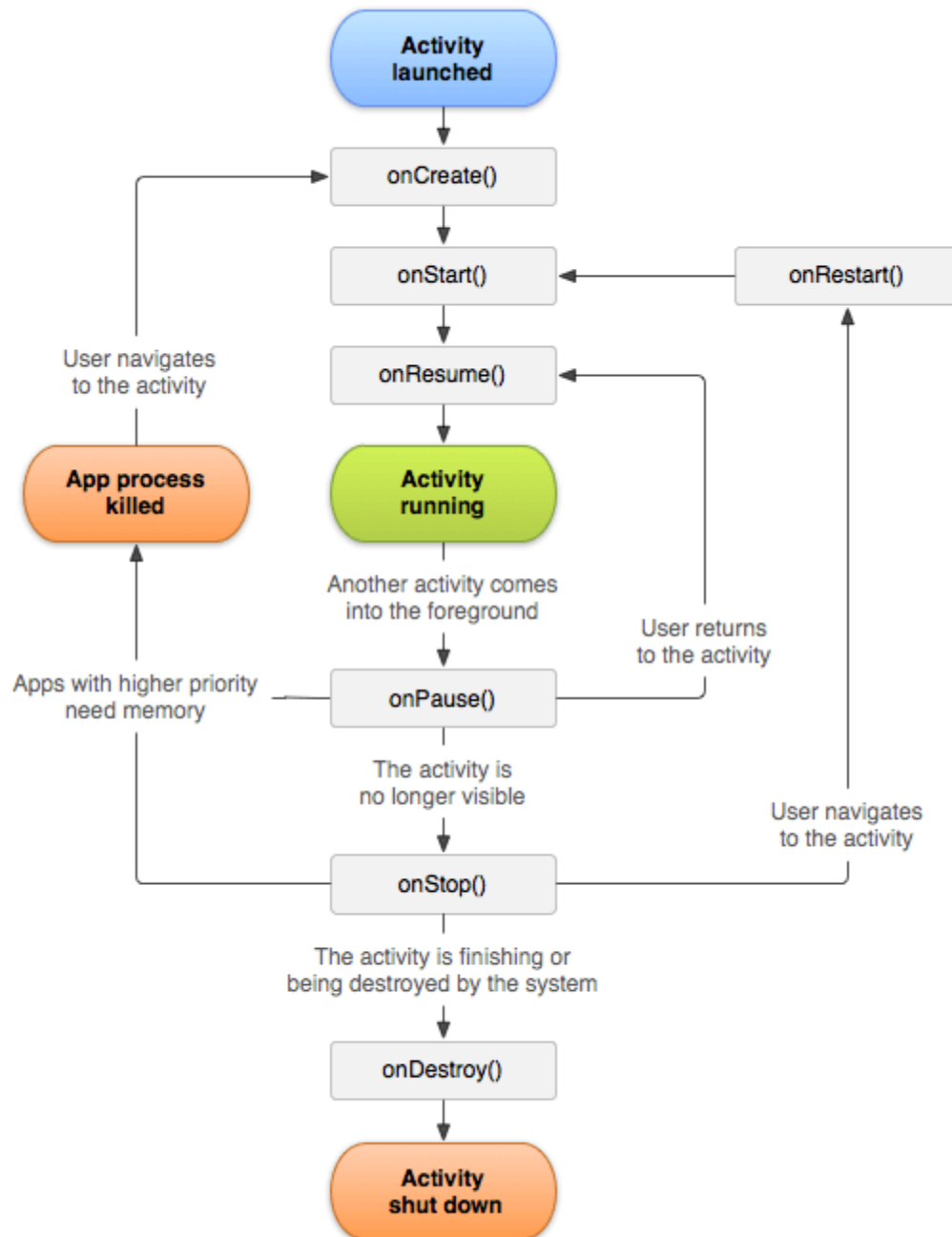
### 5.5.1.1 الفعاليات Activities

وهي تشكل شاشة وحيدة screen مع واجهة تخاطب المستخدم، مثلاً تطبيق البريد الإلكتروني يحتوي على فعالية يتم من خلالها إظهار قائمة بالرسائل الجديدة أو قراءة الرسائل مثلاً... ورغم أن هذه الفعاليات تشترك جميعها لتشكيل بنية التطبيق إلا أن كل منها يعمل بشكل مستقل عن الآخر، كما يمكن لتطبيق ما أن يقوم ببدء أي من الفعاليات بحيث يمكن لتطبيق الكاميرا أن يبدأ بفعالية تشكيل بريد إلكتروني جديد مثلاً.

تعتبر الفعالية التي تظهر في بداية إقلاع التطبيق الفعالية الرئيسة main ومن ثم يمكن لكل فعالية استدعاء فعالية أخرى وهكذا من أجل تنفيذ المهام المختلفة وعندما تبدأ فعالية ما تنتهي أو تتوقف الفعالية السابقة ويتم وضعها في مكس LIFO يدعى Back stack ولاحقاً عندما يضغط المستخدم على زر العودة Back يتم إغلاق الفعالية الأخيرة التي تم تشغيلها ومتابعة آخر فعالية تم وضعها في المكس.

يتم التعامل مع الفعالية كصف جزئي SubClass من الصف Activity ويتم فيه تعريف الطرق التي سيستدعيها النظام عن انتقال الفعالية بين المراحل المتعددة لدورة حياتها onPause و onCreate مثلاً يمكننا أن نجعل الفعالية تبدأ عن طريق استدعاء التابع startActivity مع تمرير Intent توصف النشاط ويتم إنهاء الفعالية عن طريق استدعاء التابع finish





الشكل 5 — دورة حياة الفعالية

### 5.5.1.2 الخدمات Services

وهو مكون يعمل في الخلفية background لإنجاز العمليات طويلة الأمد أو لتنفيذ عملية لإجرائية أخرى Remote process ولا يوجد للخدمات واجهة تخاطب مع المستخدم (مثلاً تشغيل الموسيقى في الخلفية بينما يعمل المستخدم على تطبيق آخر أو تحميل البيانات عبر الإنترنت من دون مقاطعة المستخدم أثناء تفاعله مع واجهة تطبيق ما) كما ويمكن استدعاء خدمة ما service من قبل مكون لتطبيق آخر غير الذي تتبع له الخدمة.

يمكن للخدمة أن تأخذ إحدى الصيغتين التاليتين:

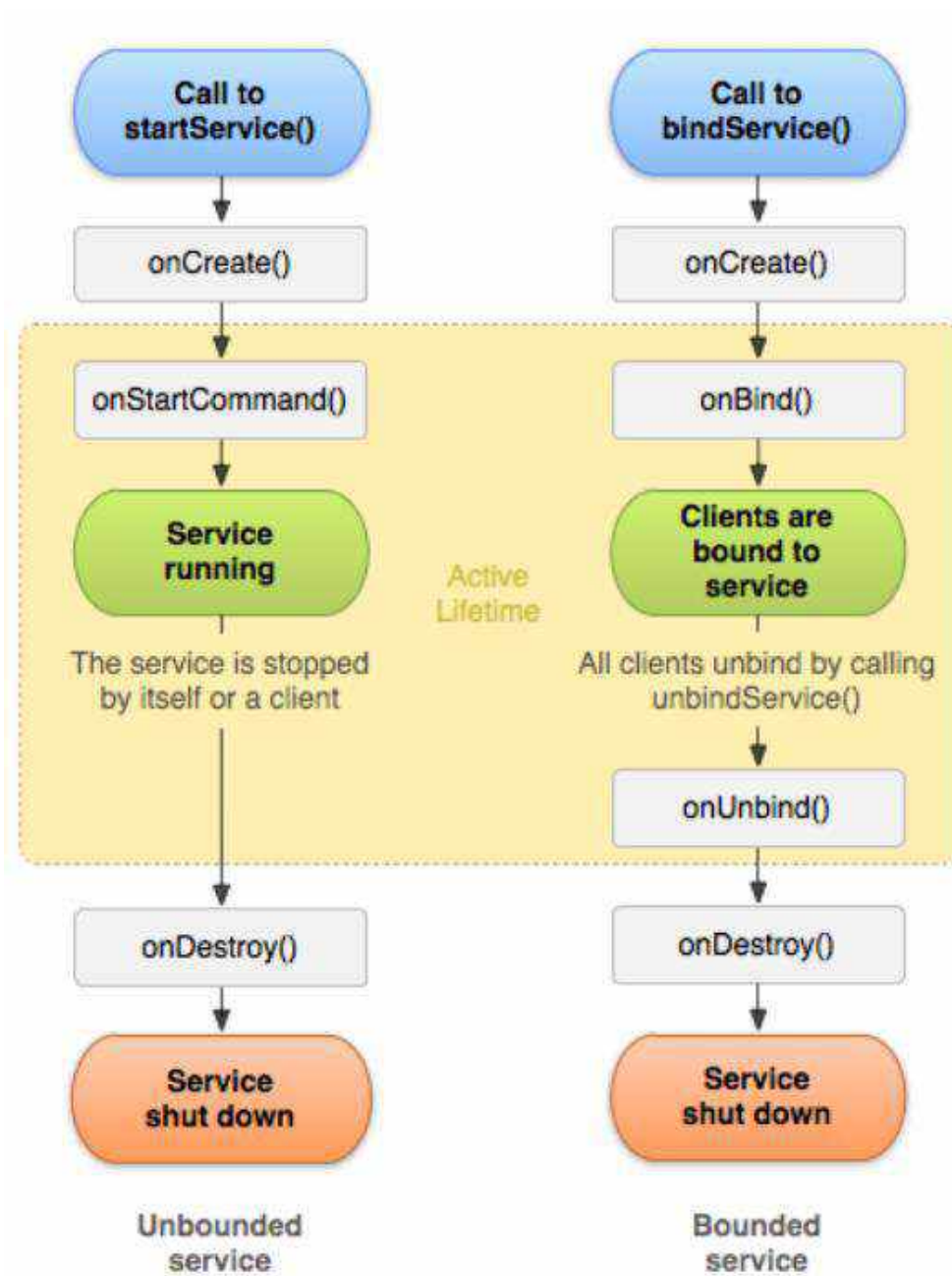
- البدء started

وهي الحالة التي تصبح عليها عندما يتم تشغيلها من قبل فعالية ما مثلاً باستخدام startService وبعدها يمكن للخدمة أن تبقى فعالة حتى بعد أن يتوقف مكون التطبيق الذي استدعاه.

- حالة الارتباط Bound

يتم ارتباط الخدمة بمكون تطبيق ما عندما يستدعي هذا المكون bindService والتي تسمح بتشكيل واجهة عميل-مخدم مما يمكن مكون التطبيق من التفاعل مع هذه الخدمة، كإرسال الطلبات إليها واستلام النتائج منها وتعمل الخدمة في هذه الحالة طالما أن المكون الذي طلب الارتباط بها يعمل ويمكن أن تكون مرتبطة بأكثر من مكون في وقت واحد.

يتم التعامل مع خدمات الصف الجزئي SubClass من الصف Service ويوضح الشكل التالي دورة حياة الخدمة:



الشكل 5 — 6 دورة حياة الخدمة

## 5.5.2 ملف Manifest

قبل بدء أي مكون من مكونات التطبيق يتعين على النظام أن يعلم بوجود هذا التطبيق عن طريق قراءة `AndroidManifest.xml` والذي يحتوي أيضا على:

- التعريف بصلاحيات المستخدم التي يحتاج إليها التطبيق (كقراءة محتويات دليل الهاتف مثلا).
- التعريف بال `Hardware` و `Software` اللازم من أجل عمل التطبيق.
- التعريف بأقل `API Level` يمكن أن يعمل عليها التطبيق، وهي رقم إصدار نظام التشغيل Android

عند تعريف كائن `Component` في ملف `Manifest File` التابع لتطبيق ما يمكن تضمين `Intent filter` تقوم بتعريف إمكانيات هذا الكائن في الاستجابة ل `Intent` من تطبيقات أخرى يمكن تعريف `Intent filter` من خلال إضافة العنصر `<intent element>` تحت تعريف الكائن المطلوب.

بعض الخواص المطلوبة في التطبيق التي يمكن مراعاتها في ملف: `Manifest`

- حجم الشاشة وكثافتها:

`extra large, large, Screen: normal.`

`extrahigh, high, medium, Density: low.`

- إعدادات الإدخال:

( `Hardware` ) `Trackball, Keyboard,` ومن النادر استخدامها في التطبيقات.

- مميزات الجهاز:

كأن يكون له الكاميرا مثلا، حساس إضاءة أو `Bluetooth...` ومن الهام جدا تعريف الإمكانيات المطلوبة من أجل عمل تطبيق ما.

## الفصل السادس

### 6 نظام تشغيل الروبوت (ROS) Robot Operating System

## 6.1 مقدمة

خلال الأعوام القليلة السابقة تقدّم موضوع الروبوتات بشكل كبير و أصبح موضع اهتمام الكثير، فبالنظر إلى بنية الروبوتات نجد أنها بشكل عام تتكوّن من مجموعة حساسات (sensor) و مشغلات (actuator) فمثلاً إذا أراد أي شخص تطوير روبوت ذو عجلتين فأول ما سيبدأ به هو تحريك الروبوت للأمام والخلف والدوران ومن ثم تفعيل الحساسات الضرورية للمهمة التي سيقوم بها الروبوت ولتجنب تكرار خطوات بناء الروبوت من الصفر تم إيجاد نظام ROS وهو منصة مستخدمة لتطوير الروبوتات (Robotic Framework) هو نظام مفتوح المصدر يمكن الاستفادة من تطوير المسبق لبعض الخوارزميات وتأمين منصة تواصل بين مختلف العتاديات.



### الشكل 6 — 1 شعار ROS

**ROS** هو نظام ولكنه ليس نظام تشغيل كنظام windows وإنما يحتاج إلى نظام تشغيل ليعمل عليه مثل OS-X & apple Linux وغيرهم وتحديداً توزيعه Ubuntu وذلك بسبب الدعم (الرسمي) الحالي له . لذا نُطلق على نظام الروبوتات هذا بـ meta operating system أي نظام تشغيل يعمل بوجود نظام آخر.

الميزة الأساسية التي تمتع بها ROS أنه صمم ليسهل إعادة استخدام المكتبات بين الروبوتات المختلفة بطريقة سهلة وركز أكثر على توفير البنية الأساسية لتطوير الروبوتات.

## 6.2 تاريخ ROS وبداية تطويره

بداية تطوير ROS كانت تحت اسم Switch-yard في مختبر Stanford Artificial intelligence laboratory بعدها تحول التطوير الأساسي (البداية الفعلية لـ ROS) إلى Willow Garage

وبعدها تم تأسيس منظمة OSRF (Open Source Robotics Foundation) وهي المشرف على ROS حتى الآن ولديها مشاريع أخرى مرتبطة مثل Gazebo وهو Physical Simulator.

### 6.3 ميزات ROS الرئيسية وأهم نقاط قوته

#### • Code Reuse & ROS-agnostic Libraries

أهم نقطة في تطوير الروبوتات أن تتم إعادة استخدام الأكواد بطريقة سهلة وسريعة. يعالج ROS هذه النقطة بحيث يكون تطوير أي برنامج مصمم أصلاً للعمل على ROS بالتالي يسهل استخدامه مع أي روبوت يعمل به أو يتم بناء برنامج wrapper للتواصل بين ROS و البرامج الأخرى لتسهيل التعامل معها.

#### • لغات البرمجة المدعومة

ROS مصمم بحيث ألا يكون مرتبط أو متوقف على أي لغة برمجة معينة. يمكن ببساطة إضافة أي لغة جديدة للعمل معه في أي وقت. أغلب الأكواد الآن مكتوبة بـ python و ++C ولكن هناك العديد من اللغات المدعومة مثل Java و Ruby وغيرها.

#### • الحوسبة الموزعة distributed processing

كل برنامج في ROS يعمل في إجرائية (Process) منفصلة أو عدد من إجرائيات ويربط بينهم باستخدام بنية اتصالات ROS الأساسية (ROS communication infrastructure). لا يهم إذا كانت هذه الإجرائيات على نفس الجهاز أم لا فـ ROS مصمم ليعمل على أي عدد من الأجهزة معاً. يمكن أن يكون لديك أي عدد من الأجهزة تعمل بـ Linux معاً لا يوجد مشكلة في ذلك. ويتم ذلك بطريقة سهلة دون التأثير على باقي البرامج الموجودة على أي جهاز منهم.

#### • Scaling

ROS قادر على تشغيل أي روبوت سواء كان معقد جداً أو بسيط جداً. حيث أن متطلبات تشغيل ROS تكبر أو تصغر حسب تطبيق الروبوت.

- **سهولة بناء الحزم (Packages) في ROS**  
يعتمد ROS على مجموعة من الأدوات لتسهيل بناء وإدارة الحزم بداخله لتسهيل نقلها بين أنظمة التشغيل وهي أدوات cmake.

- **سهولة استخدام الحساسات (sensors)**  
يوفر ROS مجموعة كبيرة جداً من drivers. تقريباً يوجد drivers لأي Sensor يمكن أن تفكر في استخدامه نتيجة لكبر شريحة المطورين. كما يوفر تكامل سهل للحصول على البيانات من Sensors متصلة بالمتحكمات الدقيقة مثل Arduino وغيره.

- **Bag files**  
تعتبر ميزه هامة في ROS. حيث أن أي برنامج يعمل في ROS يعمل على هيئه إجرائية process وبالتالي أي Sensor يعمل في ROS له إجرائية منفصلة، يتم التواصل بين هذه الإجرائيات عن طريق رسائل Messages.  
Bag files هي أداة تستطيع تسجيل رسائل بيانات الإجرائيات messages في ROS. ويمكن تسجيل كل الرسائل في النظام، وهذا معناه انه يمكنك تسجيل كل القراءات التي تمت من أي sensor أو أي إجرائيات تمت عليها وأي خرج من أي إجرائية واستخدام هذه التسجيلات في اختبارات لاحقه وكل هذا يمكن كتابته بسطر واحد فقط في سطر أوامر لينكس (Linux cmd line). هذه الأداة توفر ساعات كبيره جدا من العمل لمطوري الروبوتات وتمكنك في بعض الأحيان من القيام بأعمال لن تقدر عليها بدونها.

- **إمكانية تخاطب العقد عبر الرسائل بين بعضها حتى لو كانت مكتوبة بلغات برمجية مختلفة.**



- يدعم عمليات المحاكاة بنوعيتها (2D & 3D Simulation) عن طريق برنامج Gazebo & Stage.

## 6.4 متطلبات العمل على ROS

### 1. سطر أوامر لينكس Linux command line.

للعمل على ROS يجب أن يكون لديك معرفه بالعمل على نظام لينكس عموماً. ROS لا يتطلب معرفه كبيره بـ Linux فالمبادئ تكفي.

### 2. لغة برمجته واحده على الأقل:

يفضل معرفه python ولكن إذا لم يكن هناك معرفه بأي لغة أخرى فلغة python سهلة التعلم.

### أي نظام بنينه في ROS سيشمل العمل على الآتي:

- العمل على sensors (حساسات) مختلفة.
- استخدام المتحكمات الدقيقة Micro-controllers المختلفة.
- التعامل مع بروتوكولات الاتصال المختلفة.
- فهم طريقته عمل أنظمه التشغيل خاصة Linux.
- بناء الأكواد والمكتبات والحزم من المصدر (source code).
- العمل على معماريات المعالجات المختلفة في المشروع الواحد.
- استخدام عده لغات برمجته.
- تقليل الاعتماد على بيئات التطوير IDE والدخول في عمليه بناء الأكواد يدوياً بشكل أكبر.

ويمكن أن يضم ما يلي:

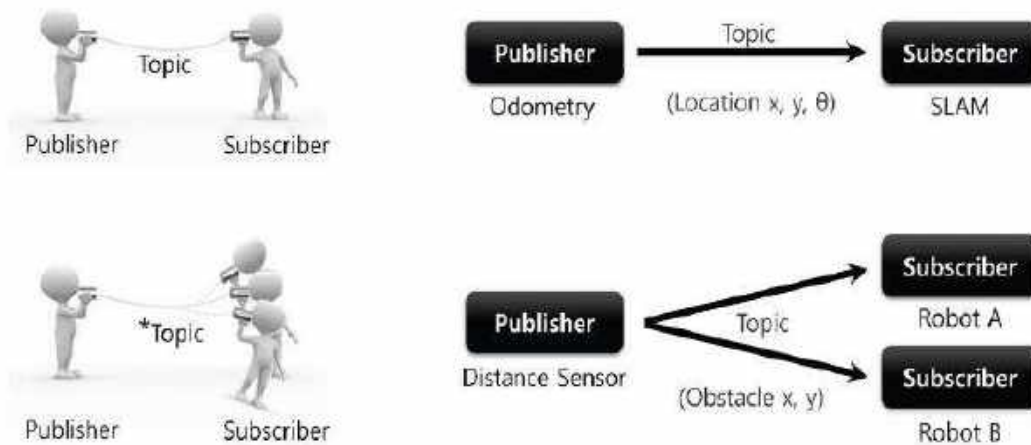
- معالجه الصور الرقمية (image processing)
- بناء الخرائط للبيئة المحيطة بالروبوت SLAM: Simultaneous Localization And Mapping
- الرؤية الحاسوبية Computer Vision وتطبيقاتها مثل:

- تمييز الأشياء. object recognition
- التعرف على الوجوه Face recognition.

## 6.5 بعض المصطلحات الأساسية الواجب معرفتها في ROS

لا يمكن البدء باستخدام نظام ROS دون فهم الأساسيات وهي تتضمن مصطلحات تتكرر طوال فترة العمل على النظام وبالتالي يجب التركيز عليها، نذكر منها:

- **Nodes** هي كل برنامج صغير يعمل داخل ROS. مثال node يمكن أن تكون برنامج خاص لقراءة بيانات sensor معين أو التواصل مع متحكم دقيق micro-controller أو التحكم في سرعه الروبوت خلال سيره ذاتياً (autonomous).
- **ROS Master** وهي node (العقدة) المسؤولة عن تنظيم الاتصال بين nodes.
- **Messages** (الرسائل) هي طريقه اتصال Nodes و تبادل البيانات بين بعضهم.
- **Topic** (قناة اتصال) وهي المكان الذي يتم نشر الرسائل عليه و يمكن أن ينشر عليها أكثر من Node أو يقرأ منها أكثر من Node في نفس الوقت وهي قناة أحادية الاتجاه. كما هو موضح بالشكل 2-6.

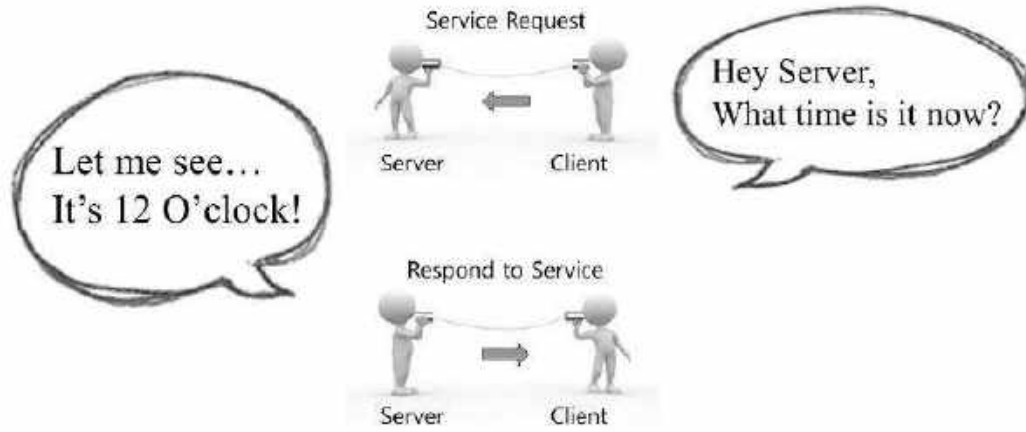


\*Topic not only allows 1:1 Publisher and Subscriber communication, but also supports 1:N, N:1 and N:N depending on the purpose.

### الشكل 6 — 2 آلية إرسال الرسائل بين العقد

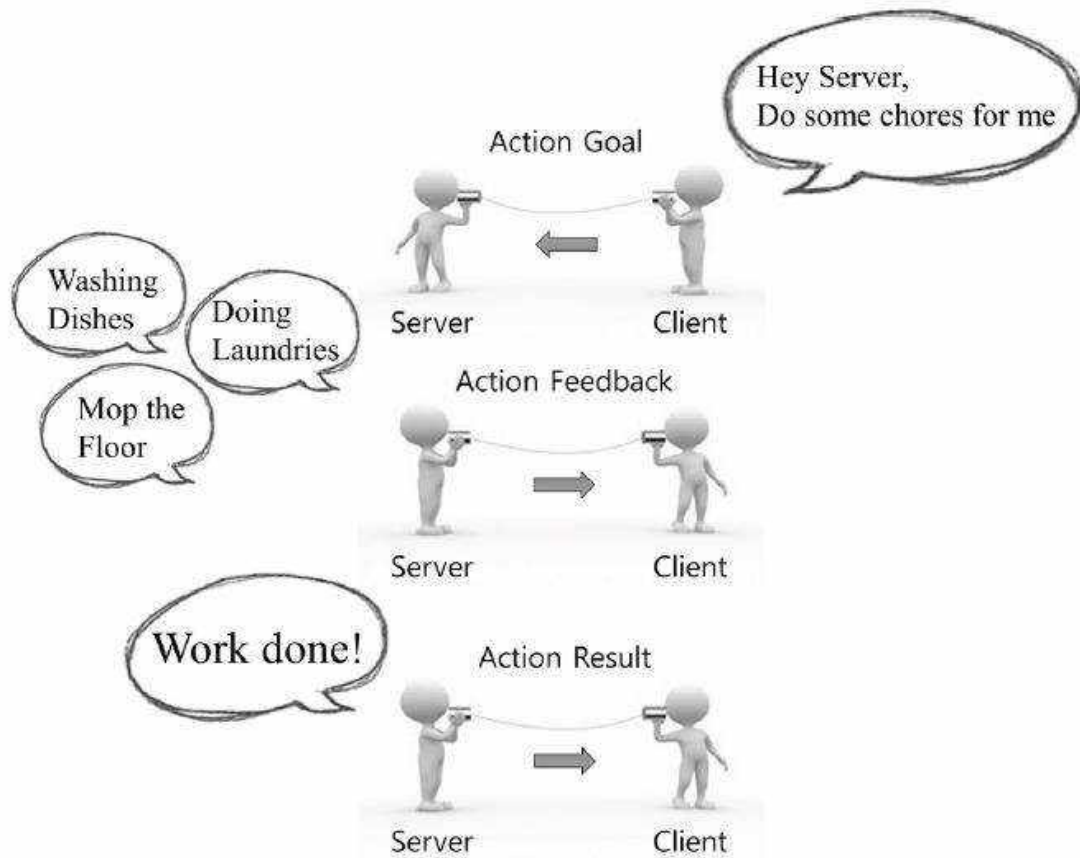
- **Publisher** هو برنامج (node) يقوم بنشر أو إرسال معطياته عبر قناة اتصال (topic) إلى برنامج أو عدة برامج أخرى.

- **Subscriber (المستقبل)** هو البرنامج الذي يراقب قناة الاتصال ويستقبل المعطيات منها.
- **Service** طلب الخدمة يكون ثنائي الاتجاه عندما يطلب برنامج client خدمة من برنامج آخر server فإنه ينتظر الرد عليه وتبقى القناة مشغولة طول فترة معالجة الطلب إلى حين الانتهاء، وخلال فترة المعالجة لا يتم إرسال أي معلومات عن حالة الطلب. كما هو موضح بالشكل 3-6



الشكل 6 — 3 آلية التواصل بين مخدم وعميل

- **Action** شبيهة بال service ويكمن الفرق بينهما بأنه في هذه الحالة يتم إرسال معلومات كتغذية راجعة (feedback) تخبرنا عن حالة الطلب إلى حين الانتهاء من تنفيذ الخدمة المطلوبة. كما هو موضح بالشكل 4-6 .



الشكل 6— Action بين مخدم وعميل في ROS

- **Bag** و يتم عليها تسجيل messages التي نريدها (تم شرحها سابقاً).
- **rostopics** هي مجموعة أدوات من Command Line يوفرها ROS للعمل عليه.
- **Ros package** تتألف الحزمة من عقدة واحدة أو مجموعة من العقد ومجموعة الحزم تشكل ما يسمى stack package.

## 6.6 ترتيب إصدارات ROS من الأحدث للأقدم

هناك بعض الإصدارات التي يمتد الدعم لها لحوالي 5 سنوات وهي إصدارات **LTS: Long Term Support** وهم حتى الآن إصدارين Indigo Igloo و Kinetic Kame.

- **Melodic**
- **Lunar**
- **Kinetic Kame**

- Jade turtle
- Indigo Igloo

Distro	Release Date	Poster	Symbol	EOL Date
Lunar Loggerhead	2017.05.23			2019.05
Kinetic Kame (Recommended)	2016.05.23			2021.04 (Xenial EOL)
Jade Turtle	2015.05.23			2017.05
Indigo Igloo	2014.07.22			2019.04 (Trusty EOL)

الشكل 6 — 5 إصدارات ROS

البقية انتهى الدعم لهم:

- Hydro Medusa
- Groovy Galapagos
- Fuerte Turtle
- Electric Emys
- Diamondback
- C Turtle
- Box Turtle

يوجد توافق بين إصدارات Ubuntu وإصدارات ROS فمثلاً إصدار Indigo LTS يعمل مع Ubuntu 14.14 LTS (Trusty Tahr) وإصدار kinetic Kame يعمل مع Ubuntu 16.04 LTS (Xenial Xerus).

## 6.7 تثبيت ROS (indigo)

نظام التشغيل المطلوب: Ubuntu 14.04/13.10

### 6.7.1 خطوات تنصيب نظام Ubuntu 14.04 LTS

1. نقوم بتنزيل نسخة من نظام التشغيل Ubuntu من الموقع الرسمي ونختار النسخة التي لها دعم طويل الأمد LTS.
2. بعد تنزيل النظام نقوم بتنصيبه على flash memory يكون حجمها على الأقل 4 gigabyte، وللقيام بهذه المهمة نحتاج تنزيل برنامج اسمه Universal USB Installer مبين بالشكل 6-6



الشكل 6-6 إعدادات تنصيب على Flash memory

3. من واجهة البرنامج نلاحظ أنه مقسم إلى خطوات:
- (1) Step1: نحدد إصدار النظام وفي حالتنا هو Ubuntu.
  - (2) Step2: نختار نظام التشغيل الذي قمنا بتنزيله.
  - (3) Step3: نختار اسم الـ flash memory مع نوع التهيئة (format).

4) Step4: تحديد مساحة لتخزين الملفات الخاصة عند العمل على flash memory. بعد الانتهاء من حرق النظام على flash memory نقوم بإعادة تشغيل النظام وضبط اعدادات الـ BIOS للإقلاع من flash memory حيث تظهر واجهة سطح المكتب لنظام Ubuntu يمكن عندها تنزيل النظام على أحد الأقراص كنظام فرعي أو العمل على الـ flash memory.

## 6.7.2 خطوات تثبيت ROS

```
1)sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list
2)sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key
421C365BD9FF1F717815A3895523BAEEB01FA116
3)sudo apt-get update
4)sudo apt-get install ros-indigo-desktop-full
5)sudo rosdep init
6)rosdep update
7)echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc
8)source ~/.bashrc
9)sudo apt-get install python-rosinstall
```

بعد تثبيت ROS يجب تهيئة مجلد العمل catkin\_ws الذي يتم بداخله بناء الحزم البرمجية، يمكن تطبيق التعليمات التالية:

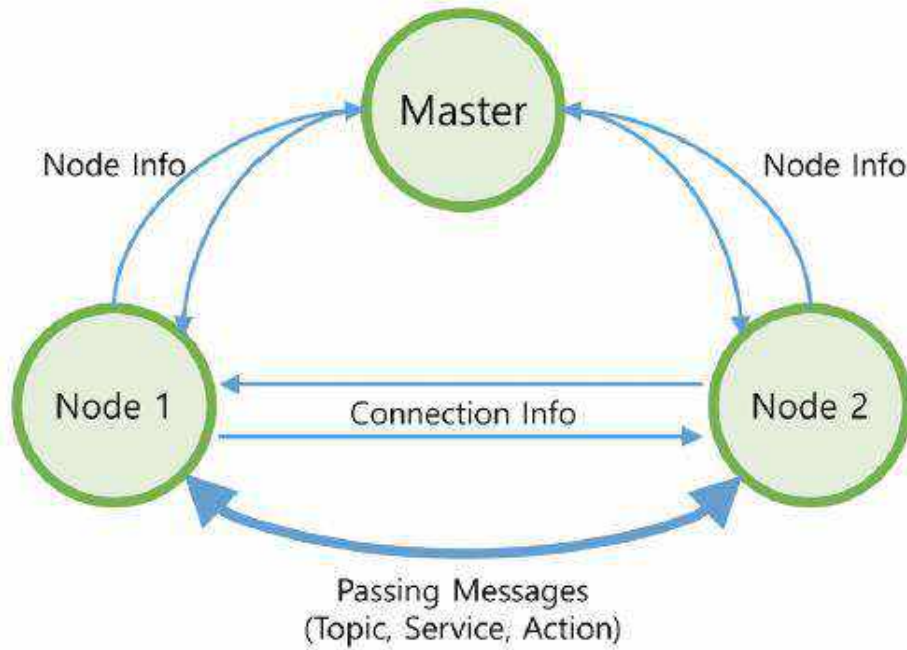
```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/src
$ catkin_init_workspace
$ cd ~/catkin_ws/
$ catkin_make
$ source /opt/ros/kinetic/setup.bash
$ source ~/catkin_ws/devel/setup.bash
```

يجب أن يحتوي مجلد catkin\_ws على ثلاث مجلدات هي (build, devel,src)



## 6.8 كيف يعمل نظام ROS

يبدأ النظام بالعمل بتشغيل عقدة رئيسية ROS Master عن طريق كتابة الأمر `roscore` في سطر الأوامر حيث تسمح هذه العقدة لبقية العقد بالتواصل والتخاطب مع بعضها البعض. إن كل مهمة أساسية تعمل في ROS تُعَدُّ عقدة (node) وتتواصل العقد مع بعضها بواسطة رسائل (messages) من المعلومات فتقوم أداة تدعى Bag file بتسجيل هذه الرسائل. تكون كل عقدة من العقد مسؤولة عن مهمة معينة، تتواصل هذه العقد مع بعضها بواسطة قنوات اتصال تدعى topics وهو المكان الذي يتم نشر رسائل المعلومات فيه. كل عقدة تخبر العقدة الرئيسية بالمعلومات (الرسائل) التي تقوم بنشرها وماهي المعلومات التي تحتاجها من العقد الأخرى فتقوم العقدة الرئيسية master node بتأمين عملية التواصل بين العقد وإنشاء قنوات اتصال فيما بينها. يبين الشكل 6-7 عملية التواصل بين عقدتين.



الشكل 6 — 7 آلية تواصل بين العقد في ROS



## 6.9 الحساسات التي يتعامل معها نظام ROS

تلعب الحساسات دوراً أساسياً في عالم الروبوتات، حيث تقوم بجمع المعلومات من البيئة المحيطة فيتم معالجتها ثم إرسالها للروبوت ليصبح قادر على أداء المهام الموكلة إليه. يدعم نظام ROS أكثر من 90 روبوت مختلف وأكثر من 80 حساس وخيارات دعم حساسات أكثر هو أمر سهل فهو نظام مفتوح المصدر (open source).



الشكل 6 — 8 حساسات ممكن استخدامها في ROS

## 6.10 المتحكمات الممكن استخدامها مع نظام ROS

يدعم نظام ROS العديد من المتحكمات واللوحات التطويرية التي تقوم بعملية معالجة البيانات القادمة من الحساسات، حيث تختلف الروبوتات عن بعضها البعض في أنواع المتحكمات المستخدمة، ففي الروبوتات البسيطة قد لا نحتاج إلى متحكمات ذات قوة معالجة كبيرة وفي بعض الروبوتات قد نحتاج إلى متحكمات أو معالجات ذات كفاءة عالية، والبعض الآخر يقوم بتحصيل البيانات من الحساسات وإرسالها إلى الحاسب ليقوم بعملية المعالجة ثم إعادة نتيجة المعالجة.

نذكر بعض أنواع المتحكمات واللوحات التطويرية:

### Arduino: (1

هي واحدة من أكثر اللوحات التطويرية انتشاراً تتميز بأنها مفتوحة المصدر، سهلة البرمجة، رخيصة الثمن مما كتب لها النجاح الكبير. معظم لوحات (arduino) مجهزة بمتحكمات صغيرة من شركة Atmel بدءاً من 8 bit وحتى 32 bit وبتردد من 8 MHz وحتى 84 MHz.

الوظيفة الرئيسية لـ arduino في الروبوتات هي ربط الحساسات (sensors) والمشغلات (actuators)، وتأمين الاتصال مع الحاسب لاستقبال الأوامر منه وإرسال قيم الحساسات إليه عبر بروتوكول UART.

يوجد أنواع مختلفة من لوحات Arduino متاحة في السوق ويعتمد الاختيار فيما بينها حسب التطبيق الذي نريد بناؤه، ويبين الشكل التالي بعض أنواع arduino.



Beginner : Arduino UNO



Intermediate : Arduino Mega



Advanced : Arduino Due

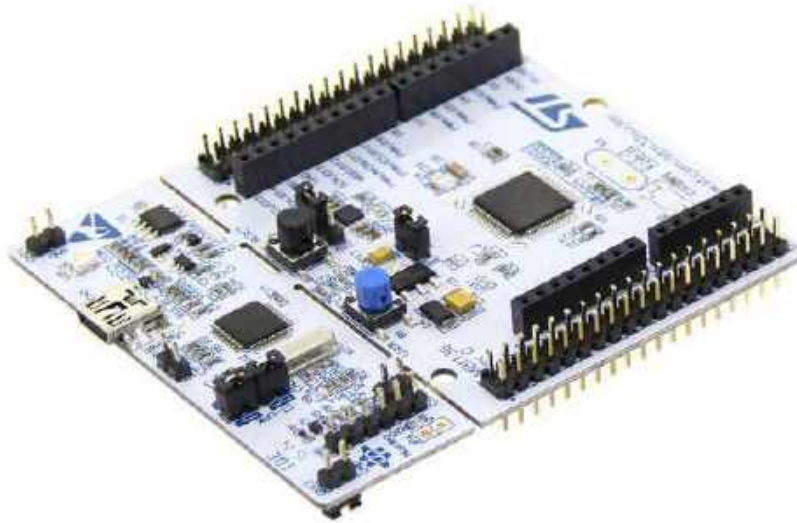
### الشكل 6 — 9 أنواع Arduino

يبين الجدول التالي بعض الخصائص العامة لكل من (Arduino Uno/Mega 2560/Due)

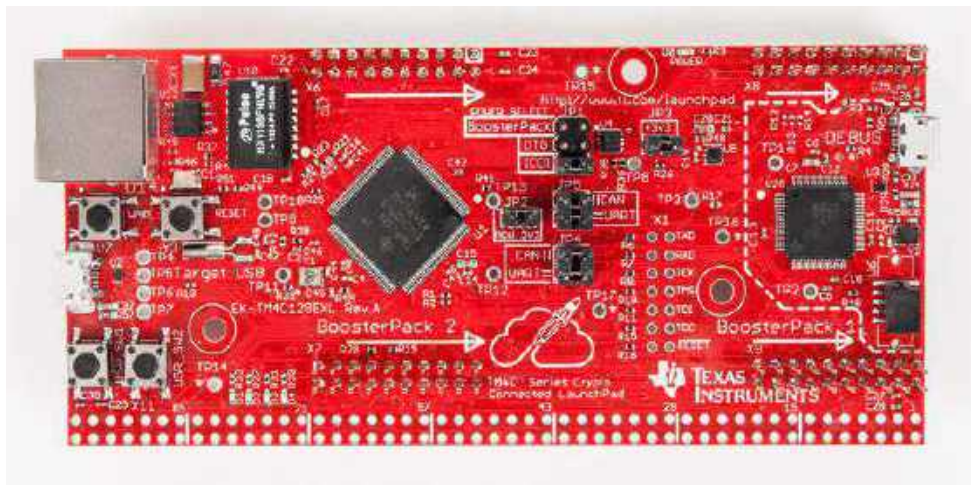
**الجدول 6 — مواصفات لوحة Arduino**

Arduino Due	Arduino Mega	Arduino	Boards
ATSAM3X8E	ATmega2650	ATmega328P	<b>Processor</b>
3.3V / 7 - 12 V	5V / 7-12 V	5V / 7-12 V	<b>Operating/Input Voltage</b>
84 MHz	16 MHz	16 MHz	<b>CPU Speed</b>
12/2	16/0	6/0	<b>Analog In/Out</b>
54/12	54/15	14/6	<b>Digital IO/PWM</b>
-	4	1	<b>EEPROM[KB]</b>
96	8	2	<b>SRAM [KB]</b>
512	256	32	<b>Flash [KB]</b>
2 Micro	Regular	regular	<b>USB</b>
4	4	1	<b>UART</b>
High end robotics application	Intermediate robotic application level application	Basic robotics and sensor interfacing	<b>Application</b>

## STM32 and TI Launchpads (2



الشكل 6— STM32 10



الشكل 6— TI Lanchpads 11

## Raspberry Pi: (3



الشكل 6 — 12 لوحة Raspberry Pi

The Odroid board (4



ODROID XU4



ODROID C2



ODROID C1+

الشكل 6 — 13 أنواع ODRIOD



## 6.11 برامج المحاكاة في ROS

من أشهر وأكثر البرامج المستخدمة في المحاكاة وتمثيل المعطيات هو RVIZ (3D Visualization Tool) و (Gazebo).

### Gazebo 6.11.1



الشكل 6 — 14 شعار GAZEBO

هو محاكي يدعم أصناف مختلفة من الروبوتات والحساسات وهو محاكي مجاني مفتوح المصدر يتميز بأدائه العالي، يستخدم بشكل واسع في مجال الروبوتيك حيث تم اعتباره المحاكى الرسمي أو المعتمد من قبل وكالة الأبحاث والمشاريع المتطورة DARPA ROBOTICS.

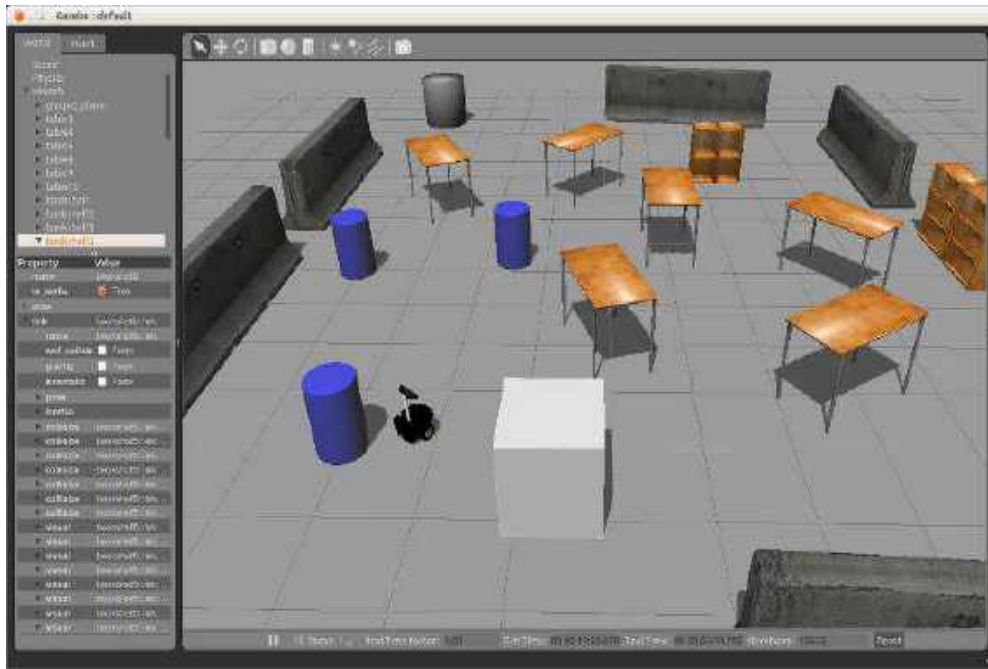
يمكن الحصول على قيم الحساسات من خلال قنوات الاتصال (Topics) و parameters و Services.

آخر نسخة لعام 2017 هي (gazebo 8.0) تعمل على نظام Ros Kinetic.

بعض الخصائص التي يتمتع بها gazebo :

- **3D Graphics** يستخدم gazebo تقنية تسمى OGRE (Open-source Graphics Rendering Engine) والتي تستخدم غالباً في تصميم الألعاب حيث يوفر تصميم أو نموذج الروبوت وبعض الإضافات مثل الظل والإضاءة والنصوص بحيث يكون المظهر واقعي قدر الإمكان.

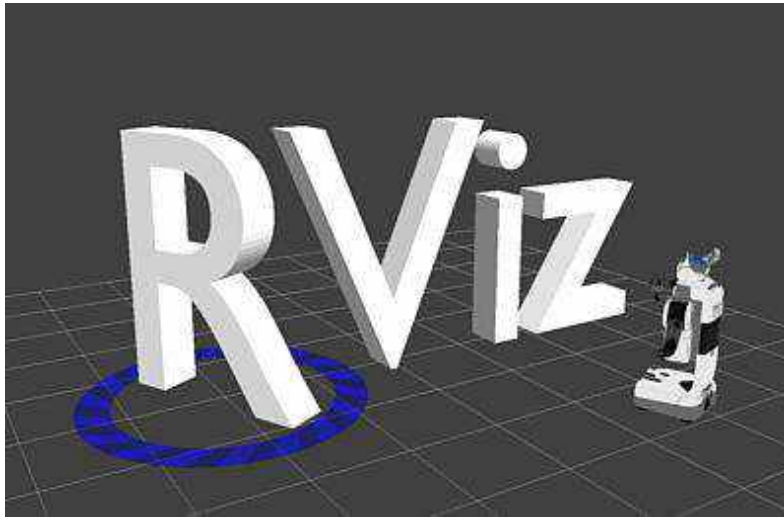
- **Sensor and Noise Simulation** يدعم أنواع مختلفة من الحساسات مثل (حساس الليزر Laser Range Finder، الكاميرات بأنواعها، حساسات التلامس أو نهاية الشوط، حساسات الضغط أو العزم، وأنواع أخرى). كما يدعم إضافة بعض الضجيج على الحساسات بحيث تصبح قريبة من الواقع.
- **البرامج المساعدة Plug-ins** تطبيقات وواجهات تسمح للمستخدم بإنشاء الروبوتات وإضافة الحساسات وتجهيز البيئة المحيطة حسب الرغبة.
- **Robot Model** بعض نماذج الروبوتات متوفرة بشكل مسبق مثل روبوت PR2 و Pioneer2 DX، iRobot Create، Turtlebot، ويمكن للمستخدم إنشاء وإضافة نماذج روبوتات أخرى.
- **Command Line Tool** يوجد أداتين هما GUI و CLI يمكن من خلالهما التحكم بحالة المحاكاة من تعديل وإضافة وغيرها.



الشكل 6—15 احدى بيئات Gazebo

### 6.11.2 برنامج التمثيل الرسومي (RVIZ) 3D Visualization Tool

هو أداة تمثيل معطيات ثلاثية الأبعاد، الهدف الأساسي منه هو رؤية الرسائل والمعطيات وتمثيلها بشكل 3D، على سبيل المثال يمكن حساب وتمثيل المسافة المأخوذة من حساس الليزر، وإظهار المعطيات المأخوذة من الكاميرا ( point cloud data)، وحساسات أخرى.



الشكل 6 — 16 شعار برنامج RVIZ

### 6.12 بعض التفاصيل عن عمل تطبيق Android يعمل كعقدة في نظام

#### ROS

أولا يجب تنصيب برنامج Android Studio على نظام لينكس وبناء بيئة تطوير Java خاصة ب ROS وإنشاء بيئة عمل خاصة بتطبيقات Android ROS يمكننا من خلالها إنشاء تطبيقات Android تستطيع أن تعمل كعقدة بنظام ROS حيث تتوفر المكتبات اللازمة الخاصة ب **rojava** وبالتالي يمكن عمل تطبيق في هذه البيئة ب Android Studio



## 6.12.1 تعليمات تنصيب Android Studio

```
$ sudo apt-get install openjdk-8-jdk
$ echo export PATH=${PATH}:/opt/android-sdk/tools:/opt/android-sdk/platform-tools:/opt/android-
studio/bin >> ~/.bashrc
$ echo export ANDROID_HOME=/opt/android-sdk >> ~/.bashrc
$ source ~/.bashrc
```

## 6.12.2 تعليمات تنصيب بيئة تطوير تطبيقات Android مع نظام ROS

```
$ sudo apt-get install ros-kinetic-rosjava-build-tools
```

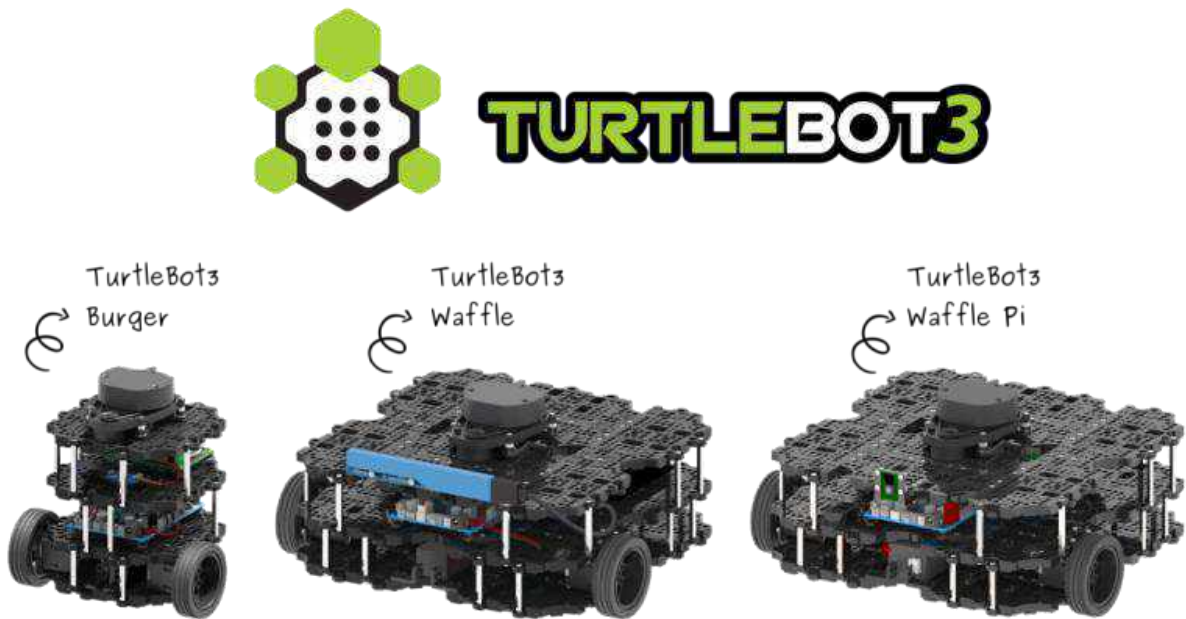
```
$ mkdir -p ~/android_core
$ wstool init -j4 ~/android_core/src https://raw.githubusercontent.com/rosjava/rosjava/kinetic/
android_core.rosinstall
$ source /opt/ros/kinetic/setup.bash
$ cd ~/android_core
$ catkin_make
```

## الفصل السابع

### 7 روبوت Turtlebot

## 7.1 مقدمة

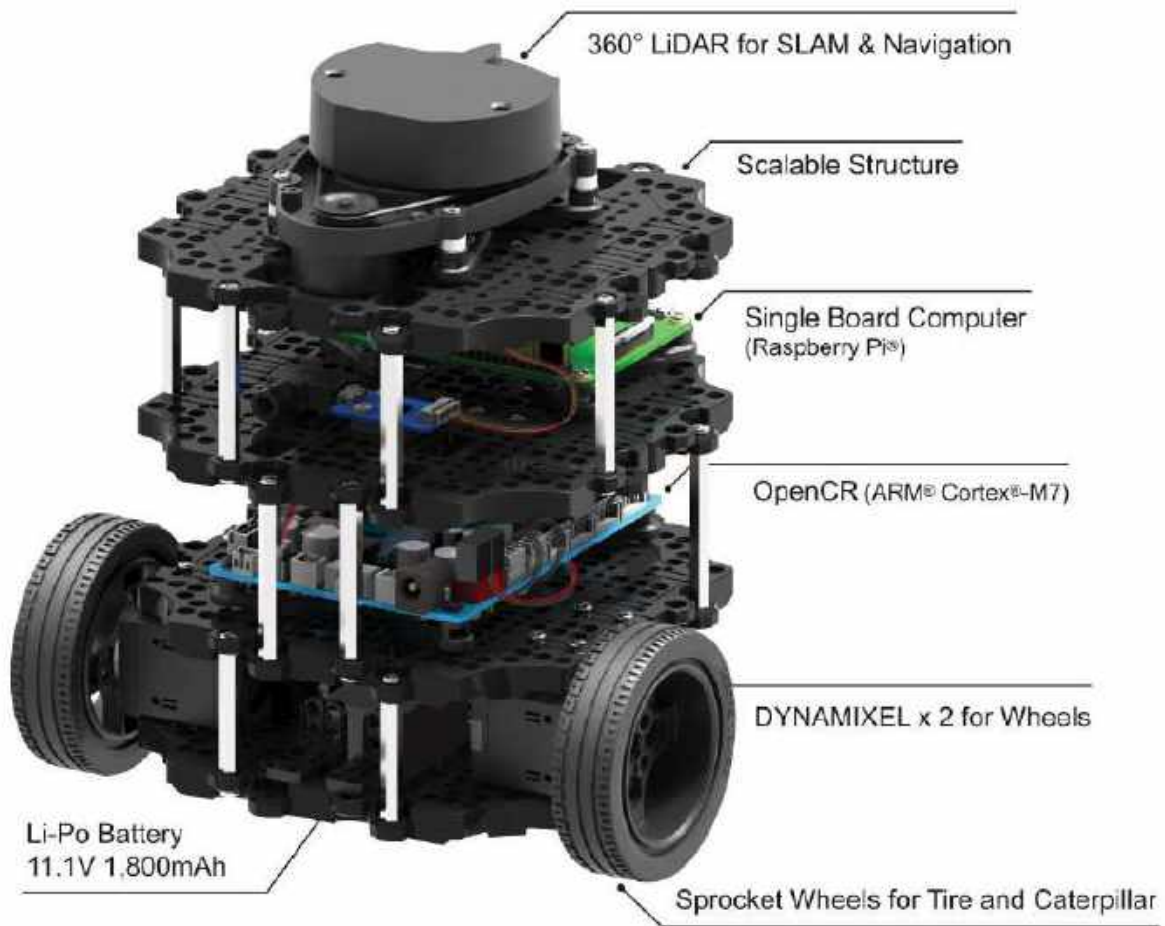
Turtlebot هي عائلة روبوتات صممت بشكل أساسي ليدعم نظام تشغيل ROS وكلمة Turtlebot هي كلمة مشتقة من Turtle robot وهو روبوت تعليمي وكان يتحكم به عن طريق لغة برمجية تعليمية Logo، عقدة turtlesim هي عقدة تظهر في الدليل التعليمي لنظام ROS وهي عبارة عن برنامج يحاكي نظام الأوامر في برنامج Logo turtle



الشكل 7 — 1 أنواع Turtlebot3

## 7.2 روبوت TurtleBot3 burger

ظهرت ثلاث سلاسل من TurtleBot أحدثها TurtleBot3 والذي صمم بحيث يحوي ميزات جديدة غير موجودة في السلسلتين السابقتين والذي يملك مشغلات شركة Robotis باسم Dynamixel للقيادة بالإضافة لاحتوائه على الحساسات الضرورية للأبحاث في علم الروبوتيك



الشكل 7 — 2 مكونات Turtlebot3 burger

### 7.2.1 حساس الليزر LDS-01

وهو حساس مسافة ليزري يدور بزاوية 360 درجة على مستوي ثنائي الأبعاد قادر على تزويد بيانات للروبوت حول البيئة المحيطة يمكن استخدامها في خوارزميات Slam و Navigation الملاحه. وهو يدعم واجهة تخاطب UART.



الشكل 7 — 3 حساس الليزر 360

### 7.2.1.1 المواصفات الأساسية

الجدول 7 — 1 مواصفات Turtlebot3

جهد التشغيل الكهربائي	5V DC $\pm$ 5%
المصدر الضوئي	ديود نصف ناقل ليزري بطول موجة nm785
معييار أمان الليزر	IEC60825-1 Class 1
استهلاك التيار	400mA
مجال المسافة	500mm, 120mm ~ 3
واجهة التخاطب	400 bps) 42bytes, 3.3V USART (230 Full Duplex option, per 6 degrees

مقاومة الضوء المحيط	10، 1000 lux أو أقل
تردد الاعتيان	1.8kHz
الأبعاد	69.5(W) X 95.5(D) X 39.5(H)mm
الكتلة	تحت 125g

## 7.2.2 اللوحة Raspberry Pi 3 Model B

تعتبر Raspberry Pi 3 Model B الجيل الثالث من إصدارات Raspberry Pi وهي عبارة عن لوحة حوسبة قوية بحجم البطاقة الشخصية ممكن استخدامها للعديد من المهام وتعتبر بالنسبة ل Turtlebot الأساس في جميع عمليات التواصل والحوسبة اذ ان عقدة bringup الموجودة ضمنها تتكفل بتشغيل واستخلاص المعلومات من لوحة OpenCR ومن الحساس الليزري وتنشرهم في Topic مناسبة حيث أن معالجها يتمكن من إنجاز كل هذه



#### الشكل 7 — Raspberry Pi 4

المهام بالإضافة إلى كونها مجهزة بشريحة للاتصال اللاسلكي WIFI مضمنة ضمن شريحة المعالجة من شركة Broadcom وبالتالي يصبح قادراً على الدخول في شبكات WIFI المحيطة بالإضافة لوجود اتصال Bluetooth. شركة Robotis أيضاً قد طرحت توزيع Linux مبنية على Debian Stretch مع بيئة سطح مكتب حيث ينصب فيها نظام ROS مع حزم Turtlebot3 الأساسية

#### 7.2.2.1 المواصفات الأساسية

##### الجدول 7 — مواصفات Raspberry Pi 3 B

وحدة المعالجة	1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
وحدة المعالجة	Dual Core VideoCore IV® Multimedia Co-Processor.

الرسومية	Provides Open GL and 1080p30 H.264 , hardware-accelerated OpenVG,ES 2.0 high-profile decode.
ذاكرة الوصول العشوائي	1GB LPDDR2
نظام التشغيل	يتم الاقلاع من نسخ من نظام Linux أو نظام Windows 10 IoT
الأبعاد	85 x 56 x 17mm
التغذية	2.5A, Micro USB socket 5V1

### 7.2.2.2 وحدات Raspberry Pi المحيطية

#### • مخرج العرض HDMI:

المخرج المستخدم في توصيل Raspberry Pi بالشاشات التلفزيونية عالية الدقة مع العلم أن هذا المخرج يقوم بإخراج كلاً من الصوت والصورة إلى شاشة التلفاز.

#### • مخرج العرض RCA video output:

يعتبر هذا المخرج مثل سابقه بالضبط إلا أنه يستخدم في توصيل Raspberry Pi مع أجهزة التلفاز القديمة.

#### • مخرج الصوت Audio jack:

مخرج الصوت بمقياس 3.5mm وهو نفس المقاس الثابت لمعظم أنواع السماعات المختلفة.

#### • مدخل كبل الشبكة Ethernet Port:

المدخل المستخدم في توصيل Raspberry Pi في شبكات الحاسب الآلي والإنترنت

#### • مداخل USB:



مثل الموجودة في أي حاسب تقليدي وتستخدم في توصيل أي جهاز USB مثل لوحة المفاتيح والماوس أو كروت الشبكة اللاسلكية ويمكن استخدام هذه المداخل في توصيل أي شيء يعمل مع الحاسب التقليدي الآلي.

- **مدخل Micro USB:**

مدخل الطاقة ويستخدم في توفير الطاقة Raspberry Pi عن طريق توصيله بأي شاحن من شواحن الهاتف التقليدية بجهد 5 Volts.

- **مدخل بطاقة الذاكرة MicroSD Card Slot**

- **GPIO PINS**

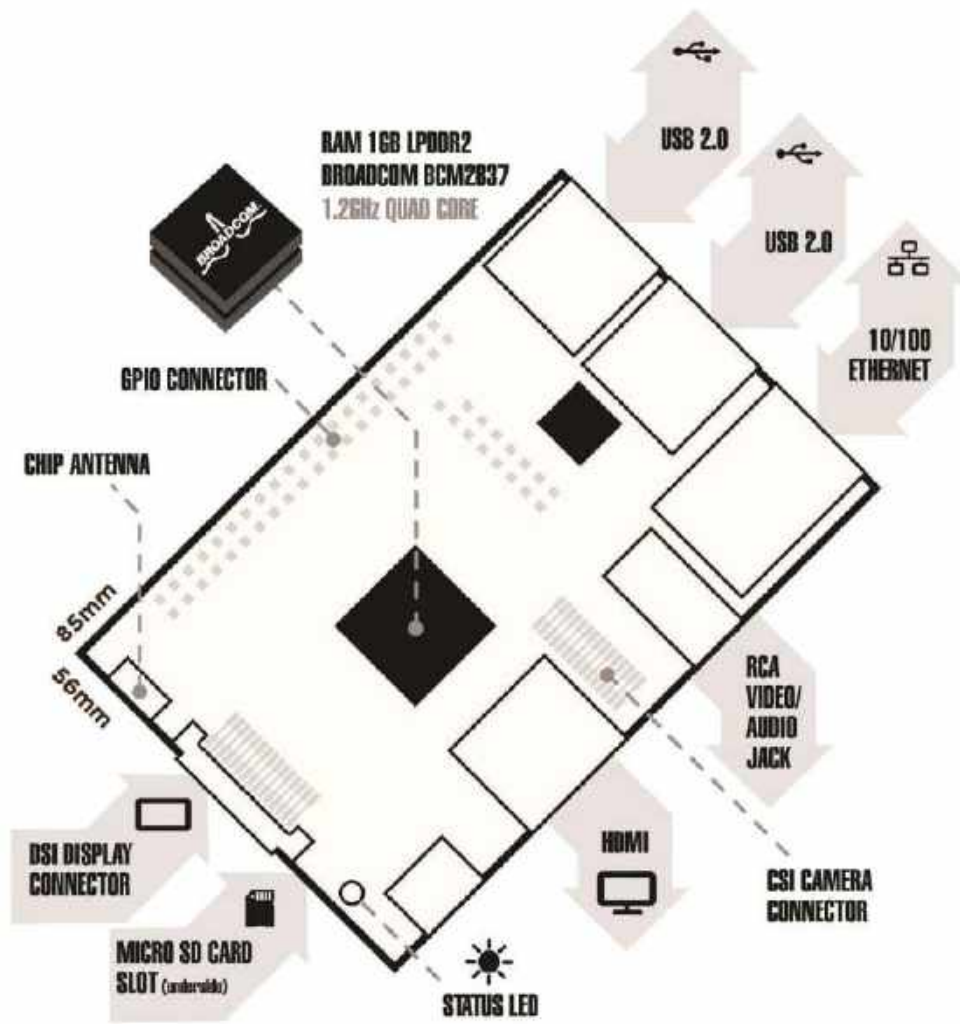
مجموعة منافذ التحكم الإلكتروني وهي المنافذ التي تمكن من التحكم في المكونات الإلكترونية المختلفة وتجعل Raspberry Pi قادر على القيام بوظائف المتحكمات الصغيرة.

- **مدخل CSI Camera Input**

مكان توصيل الكاميرات عالية الدقة والمصنعة خصيصا ل Raspberry Pi

- **مدخل DSI Display**

يستخدم هذا المدخل في توصيل شاشات اللمس مثل المستخدمة في الهواتف الذكية والأجهزة اللوحية.



الشكل 7 — 5 الوحدات المحيطية Raspberry Pi

### 7.2.2.3 التحكم في Raspberry Pi عن بعد

يمكن التحكم بالراسبيري عن طريق:

#### 1. SSH

وهي عبارة عن خدمة تفيد التحكم عن بعد لأي جهاز يعمل بنظام لينوكس عن طريق سطر الأوامر كما تتميز الخدمة بالتشفير العالي وإمكانية استخدامها في نقل الملفات وعمل اتصالات بين الأجهزة بصورة آمنة ومشفرة.

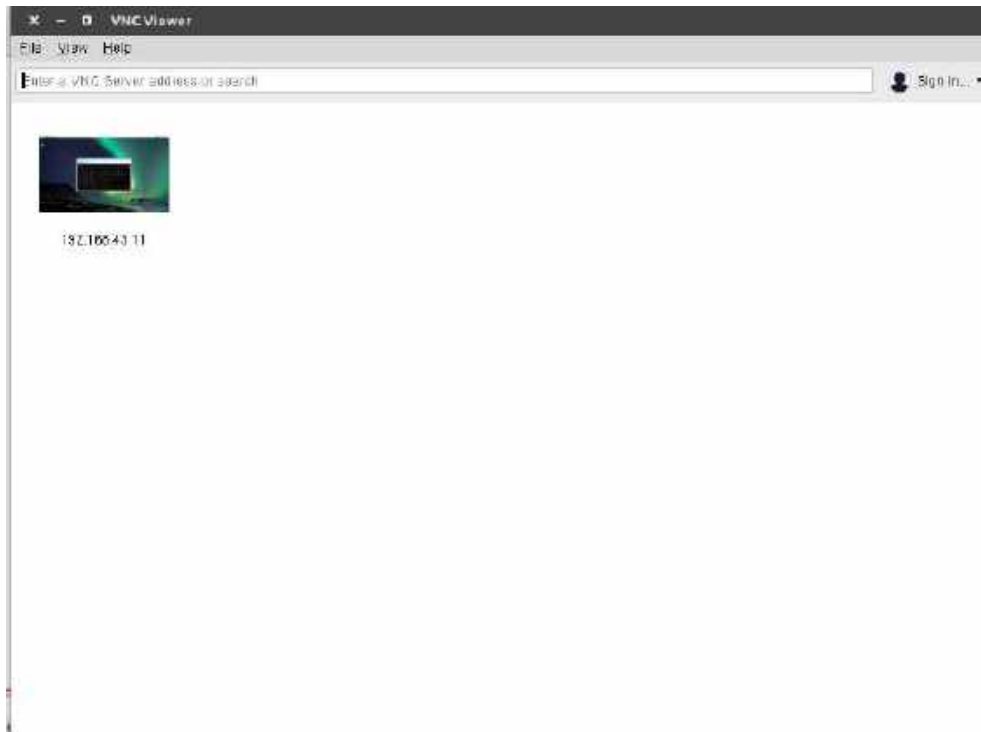
نحتاج إلى برنامج client نضعه على أي جهاز حاسب آخر مثل Putty.

## 2. VNC Server

وهو نظام الوصول إلى الأجهزة والتحكم فيها عن بعد حيث يمكننا مشاهدة سطح المكتب ونقل ما يحدث بشكل فوري، تعتمد هذه الطريقة على عمل اتصال VNC بين الجهازين عن طريق تثبيت برنامج في الجهاز الذي سيتم التحكم فيه Server وبرنامج في الجهاز الذي سيتم التحكم منه Viewer

ومن مميزات نظام الـ VNC أنه لا يعتمد على نظام تشغيل معين فمن الممكن إجراء اتصال بين جهاز يعتمد Windows وجهاز آخر يعتمد Linux.

في Raspberry Pi يتم البدء تلقائياً بـ VNC Server ومن ثم يمكننا من أي حاسوب آخر متصل ضمن شبكة Raspberry Pi أن يدخل IP الخاص بها للدخول ويفضل جعل عناوين Static IP وعدم الاعتماد على DHCP لسهولة العمل.

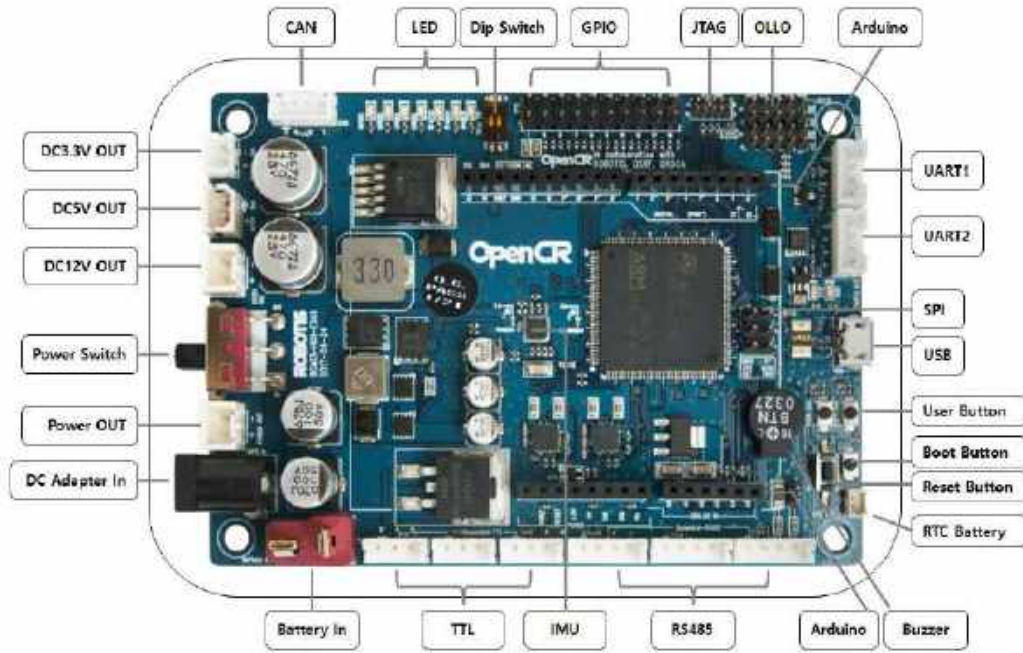


الشكل 7 — واجهة برنامج VNC Client ومكان كتابة IP الخاص بـ Raspberry Pi

### 7.2.3 دارة التحكم OpenCR

تعتبر المتحكم الرئيسي في الروبوت وOpenCR اختصار لـ

Open-source Control Module for Ros وهذه الدارة طورت لنظم ROS المضمنة لكي تعطي نظام عتادي وبرمجي مفتوح المصدر بالكامل وهي مسؤولة عن إعطاء قيم حساس IMU واعطاء أوامر التحكم للمحركات وتزويدهم بالطاقة.



الشكل 7 — 7 لوحة OpenCR

7STM32F هي الرقاقة الأساسية داخل OpenCR، مبنية على معالج قوي ARM Cortex-M7 مع وحدة معالجة الفاصلة العائمة وتحتوي حساسات التسارع والزواية و Gyroscope. بيئات التطوير مفتوحة المصدر بالكامل أيضاً هما بيئة تطوير Arduino وبيئة Scratch للمتعلمين اليافعين.

اللوحة مزودة بمداخل تماثلية وتشابيهية يمكنها أيضاً أن تبني واجهة تخاطب مع دارة إضافية أو حساسات مختلفة. كما تدعم اللوحة العديد من بروتوكولات: CAN، I2C، SPI، USB.

## 7.2.3.1 المواصفات الأساسية

الجدول 7 — مواصفات OpenCR

المتحكم	462DMIPS),STM32F746ZGT6 / 32-bit ARM Cortex® -M7 with FPU (216MHz
الحساسات	Magnetometer 3Axis (MPU9250), Accelerometer 3Axis,Gyroscope 3Axis
المبرمجة	ARM Cortex 10pin JTAG/SWD connector USB Device Firmware Upgrade (DFU) Serial
بنات التوسعة	R 18) *Arduino connectivity,32 pins (L 14 Sensor module x 4 pins Extension connector x 18 pins
دارات التخاطب	USB (Micro-B USB connector/USB 2.0/Host/Peripheral/OTG) TTL (B3B-EH-A / Dynamixel) RS485 (B4B-EH-A / Dynamixel) UART x 2 (20010WS-04) CAN (20010WS-04)
الديودات الضوئية والأزرار	LD2 (red/green) : USB communication LD5 (blue), LD4 (green),User LED x 4 : LD3 (red) User button x 2

التغذية	<p>مدخل تغذية خارجية</p> <p>7-24 V (Battery or SMPS), 5 V (USB VBUS)</p> <p>المدخلة الأساسية : 1 11.1V LI-PO 19.98Wh 800mAh</p> <p>وحدة التغذية المتقطعة الأساسية : 12V 5A</p> <p>مخرج تغذية</p> <p>3.3V@800mA(20010WS-02), 5V@4A(5267-02A), 12V@1A(SMW250-02)</p> <p>مدخل لمدخلة ساعة زمن الحقيقي (Molex) (0210-53047 RTC (Real Time Clock)</p> <p>3.3 V power on), Power LED: LD1 (red</p> <p>Reset button x 1 (for power reset of board)</p> <p>Power on/off switch x 1</p>
الأبعاد	105(W) X 75(D) mm
الكتلة	60g

## 7.2.4 محركات Dynamixel

المحركات المستخدمة هي أعلى المحركات أداءً في مجال الروبوتيك مصنعة من قبل الشركة الكورية Robotis وهي تتألف من محرك تيار مستمر متكامل و علبة سرعة ومتحكم و ومشغل برمجي و أدوات الاتصال شبكياً



الشكل 7—8 محركات Dynamixel

مواصفات المحرك Dynamixel XL-430:

الجدول 7 — 4 مواصفات محرك Dynamixel XL-430 W 250

المتحكم	32Bit)،ST CORTEX-M3 (STM32F103C8 @ 72Mhz				
حساس الموضع	مباشر	تماس	دون	من	مرمز (12Bit)، (360°)
المحرك	محرك ملفوف				
سرعة الاتصال	9600 bps ~ 4.5 Mbps				
خوارزمية	PID				

التحكم	
درجة الإحكام	0.088°
أنماط التشغيل	<p>نمط التحكم</p> <p>نمط التحكم</p> <p>نمط الإضافي للتحكم بالموضع</p> <p>نمط التحكم بنبضات PWM</p>
الوزن	57.2غ
الأبعاد	28.5mm x 46.5mm x 34mm
نسبة تحويل علبة السرعة	258.5 : 1
عزم الانطلاق	<p>1.0A, 1.0N.m @ 9.0V</p> <p>1.3A, 1.4Nm @ 11.1V</p> <p>1.4A, 1.5Nm @ 12.0V</p>
سرعة الاحمل	<p>47rpm @ 9.0V</p> <p>57rpm @ 11.1V</p> <p>61rpm @ 12.0V</p>



مجال درجة الحرارة	-5°C ~ +72°C
جهد الدخل	( 12.0V ~ 6.5V الموصى به (11.1V :
تيار الوقوف	52mA
إشارة التحكم	حزمة رقمية
نوع البروتوكول	نظام اتصال تسلسلي غير متزامن باتجاه واحد (8bit, 1stop, No Parity)
مجال الرقم المعرف	0 ~ 252
التغذية الخلفية	الموضع ، سرعة ، حمل ، الزمن الحقيقي ، المسار ، درجة الحرارة ، جهد الدخل ..
مكون التصنيع	بلاستيك صناعي مقسى

## الفصل الثامن

8 بناء نظام تحديد موقع ضمن مبنى قسم هندسة الحواسيب  
والأتمتة في جامعة دمشق

## 8.1 مقدمة

كما ذكرنا سابقاً يمكن تعميم طرق تحديد الموقع في شبكات الحساسات اللاسلكية على إشارات الشبكة الخلوية التي أيضاً تكون تتواجد ضمن المنشآت وسنستعرض في هذا الفصل كيفية بناء هذا النظام بشكل كامل ضمن بيئة داخلية Indoor في بناء هندسة الحواسيب والأتمتة في كلية الهندسة الميكانيكية في جامعة دمشق.

## 8.2 مخطط العمل

أحد الأمور الأساسية التي يجب توفرها لبناء نظام تحديد الموقع هي البيانات المعنونة، إذ تعد هذه المرحلة من أصعب المراحل والأكثر استهلاكاً للجهد والوقت ويبين الشكل 1-8 خطوات الرئيسية لبناء نظام تحديد الموقع



الشكل 8 — 1 مخطط عمل نظام تحديد الموقع

## 8.2.1 تحصيل البيانات المعنونة

كما ذكرنا سابقاً في الفصل الثالث بناء نظام تحديد الموقع بمساعدة خوارزميات تعلم الآلة يتطلب، بشكل أساسي قبل الشروع بأي شيء، معطيات معنونة، أي إحداثيات عدة مواقع في

location	Cell-1 RSSI	Cell-2 RSSI	Cell-3 RSSI	Cell-4 RSSI	Cell-5 RSSI	Cell-6 RSSI	Cell-7 RSSI
About 16 K instance:							

### الشكل 8—2 المعلومات الأساسية لبناء النظام

البيئة والإشارات الخلوية الموافقة لها.

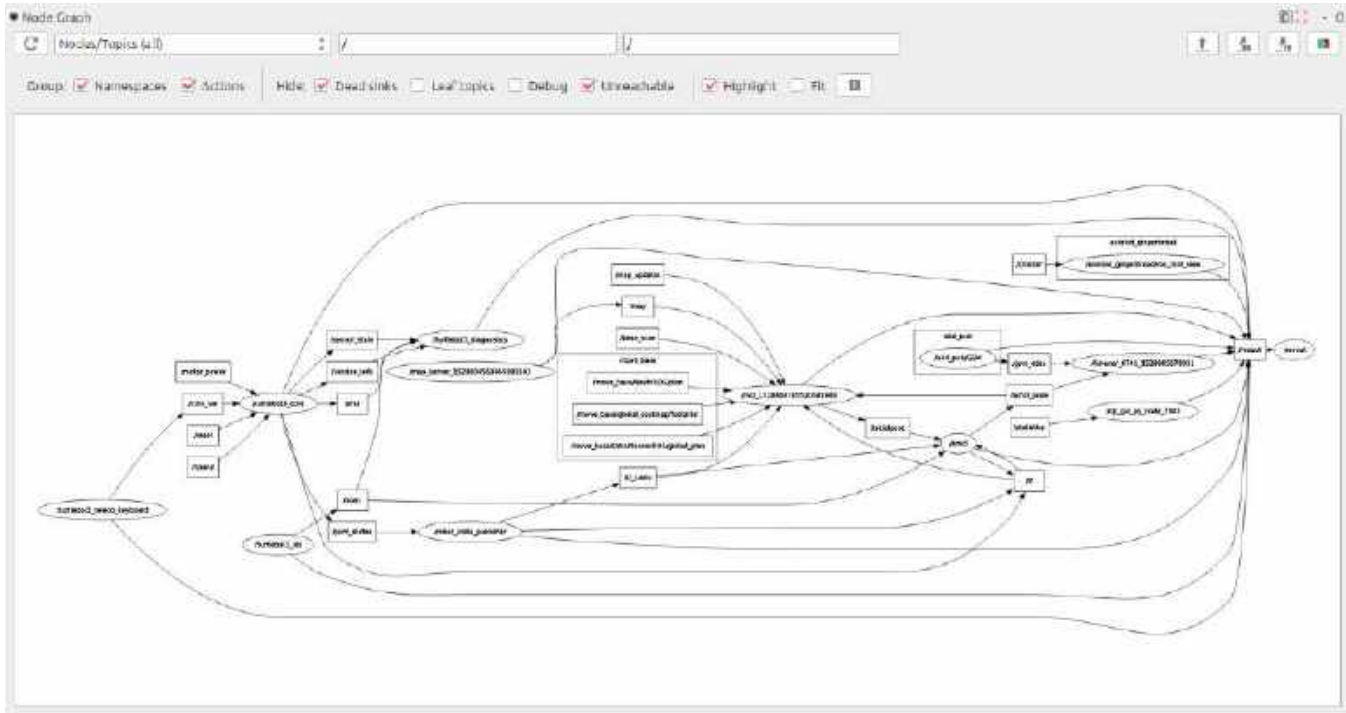
تحديد الموقع ضمن الفصل الثالث كان عن طريق الإحداثيات الجغرافية حيث أن أي عملية تحديد موقع تحتاج إلى مبدأ إحداثيات لكي ننسب هذا الموقع إليه وقررنا في مشروعنا اللجوء إلى بناء خريطة ضمن المكان الذي نريد تحديد الموقع فيه إذ تحتوي هذه الخريطة على إطار مرجعي نستطيع نسب الإحداثيات إليه. ومن أشهر خوارزميات بناء الخرائط في عالم الروبوتيك هي خوارزمية SLAM التي استخدمناها لبناء الخريطة

نحتاج أيضاً إلى تحصيل الإشارات الخلوية من شبكة GSM وهنا سنلجأ إلى استخدام تطبيق Android نستخدمه ليحصل الإشارات الخلوية وتحتاج أيضاً إلى نوع من المزامنة لتحصيل كل مجموعة إشارات خلوية بالموقع الصحيح الموافق لها وهنا سنخدم نظام ROS كي نشكل ثلاث عقد بشكل رئيسي:

1. عقدة تحصيل إشارات الخلوية (عن طريق ربط تطبيق Android بنظام ROS)

2. عقدة تحصيل الموقع الحقيقي الموافق لتلك الإشارات
3. عقدة لجمع بيانات العقد السابقة وتصدير بشكل مناسب لكي تشكل معطيات تدريب النظام.

الشكل التالي يبين مخطط rqt\_graph الذي يوضح جميع العقد وTopics المشتركة بها.

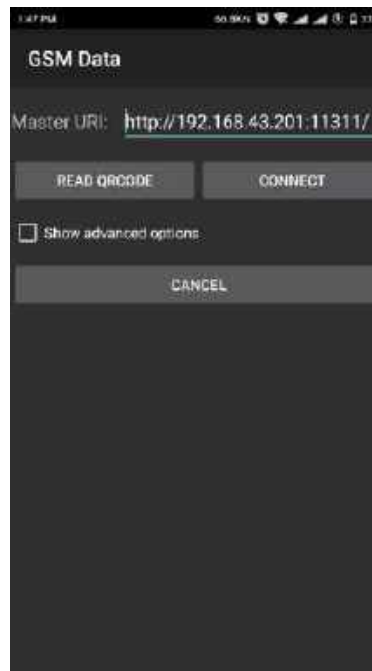


الشكل 8 — 3 مخطط rqt graph

### 8.2.1.1 مراحل استخلاص البيانات المعنونة

- بعد مراحل ربط على الشبكة ووضع IP ثابت لكل من الحاسب والروبوت
- (1) تشغيل عقدة Master على الحاسب
- (2) تشغيل عقدة bringup في Turtlebot مع ما يلزم من تعليمات للتحكم بالروبوت
- (3) تحميل الخريطة التي تم بناؤها عن طريق SLAM وتم استخدام hector mapping package وأيضا كان بالإمكان استخدام gmapping
- (4) تشغيل تطبيق Android والاتصال ب Master والتأكد من أن المعلومات تنشر على Topic الخاصة بها
- (5) تشغيل ملف launch المسؤول عن تحديد الموقع الحقيقي في ROS

- (6) تشغيل عقدة data\_formatter وهي عقدة في نظام ROS تم كتابتها في مشروع تحصيل الموقع الحقيقي وإشارات GSM الموافقة له
- (7) تطبيق Android ROS :
- اسم تطبيق GSM Data والفكرة منه إنشاء عقدة واتصالها مع Master ونشر معلومات GSM في topic يحمل الاسم gsm\_data والشكل 4-8 يبين الواجهة الرئيسية للبرنامج وطلب IP الخاص ب Master



الشكل 4—8 واجهة إدخال Master IP

#### 8.2.1.1.1 8.2.1.1.1 صلاحيات تطبيق GSM Data

- الجزء من ملف Manifest في الشكل 5-8 يوضح الصلاحيات التي يأخذها التطبيق لتحصيل معلومات عن الشبكة الخلوية والاتصال بشبكة

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android" package="org.ros.android.android_tutorial_pythub">
3
4     <uses-permission android:name="android.permission.INTERNET" />
5     <uses-permission android:name="android.permission.WAKE_LOCK" />
6     <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
7     <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
8     <uses-permission android:name="android.permission.READ_PHONE_STATE" />
9     <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
10    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
11    <uses-permission android:name="android.permission.ACCESS_COARSE_UPDATES" />
12
13    <application
14        android:icon="@mipmap/icon"
15        android:label="GSM Data"
16        android:allowBackup="true">

```

## الشكل 8 — 5 جزء من ملف Manifest

يتم التحكم بالروبوت عن طريق جهاز الحاسب بتشغيل عقدة Teleop

لكي يتم إعطاء موقع صحيح في الخريطة يتم ذلك بتحديد موقعه يدوياً لأول مرة فقط ثم تتولى ملف my\_amcl\_gsm.launch بمتابعة تغييرات مكانه ونشرها في topic مناسب عند إيقاف تشغيل العقدة data\_formatter تتم بحفظ ملف csv تخزين به جميع البيانات التي تم استحصلها على جهاز الحاسب حيث يكون أول عمودين في الملف يشكلان الإحداثيات الديكارتية في الخريطة و C تعبر عن الرقم المعرف للخلية و R شدة الإشارة الموافقة لها

	x	y	x	y	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c
1	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
2	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
3	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
4	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
5	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
6	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
7	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
8	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
9	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
10	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
11	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
12	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
13	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
14	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
15	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
16	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
17	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
18	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
19	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
20	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
21	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
22	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
23	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
24	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
25	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
26	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
27	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
28	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
29	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
30	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
31	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
32	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
33	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
34	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
35	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
36	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0
37	0.5561444848	-0.0470780231	1.2756	-0.158903	83	58333	-0.1	12902	85	11811	-0.1	12755	-0.1	53353	-0.3	0	0	0	0	0

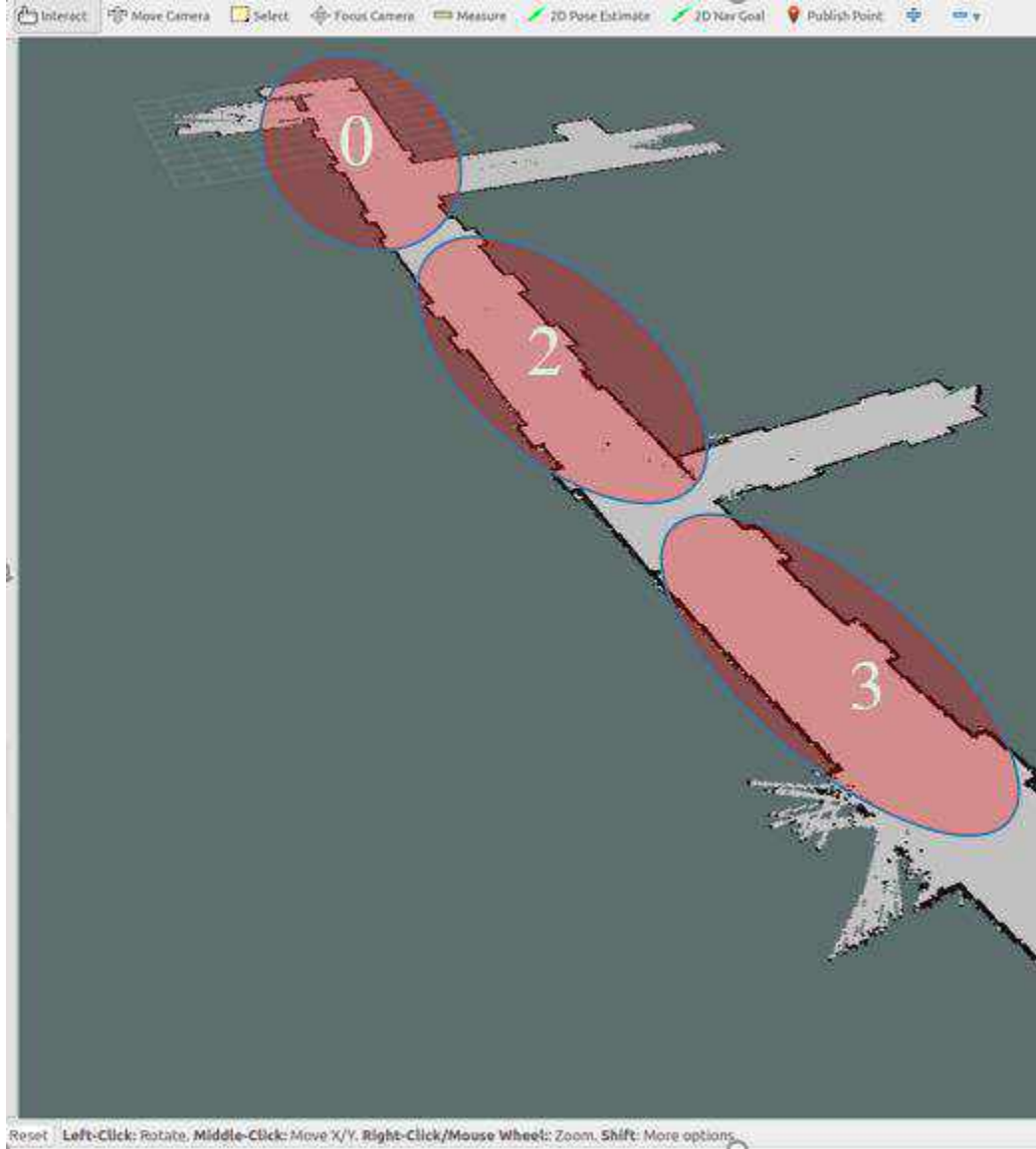
## الشكل 8 — 6 ملف csv المنشأ بعد إغلاق عقدة data\_formatter

## 8.2.2 مرحلة بناء نظام تحديد الموقع

لبناء نظام تحديد الموقع من المعطيات المحصلة يجب علينا اتباع أنبوب تعلم الآلة Machine Learning Pipeline كما في الإجراء العملي المبني على معطيات شركة الاتصالات ولكن هنا نظراً لضعف الإشارات الخلوية في بناء هندسة الحواسيب سيتم تحويل المشكلة من مشكلة انحدار إلى مشكلة تصنيف عن طريق تقسيم الخريطة إلى عدة أقسام ومن ثم نتوقع نحن بأي قسم من الخريطة بالخطوات التالية:

- معالجة المعطيات واستخلاص السمات من معطيات ملف csv
- تقسيم الخريطة إلى أربع مناطق باستخدام خوارزمية K-Means
- بناء مصنف SVM مع تقنيات kernel approximation لتصنيف الإشارات الخلوية إلى أربع مناطق مع حفظ النموذج

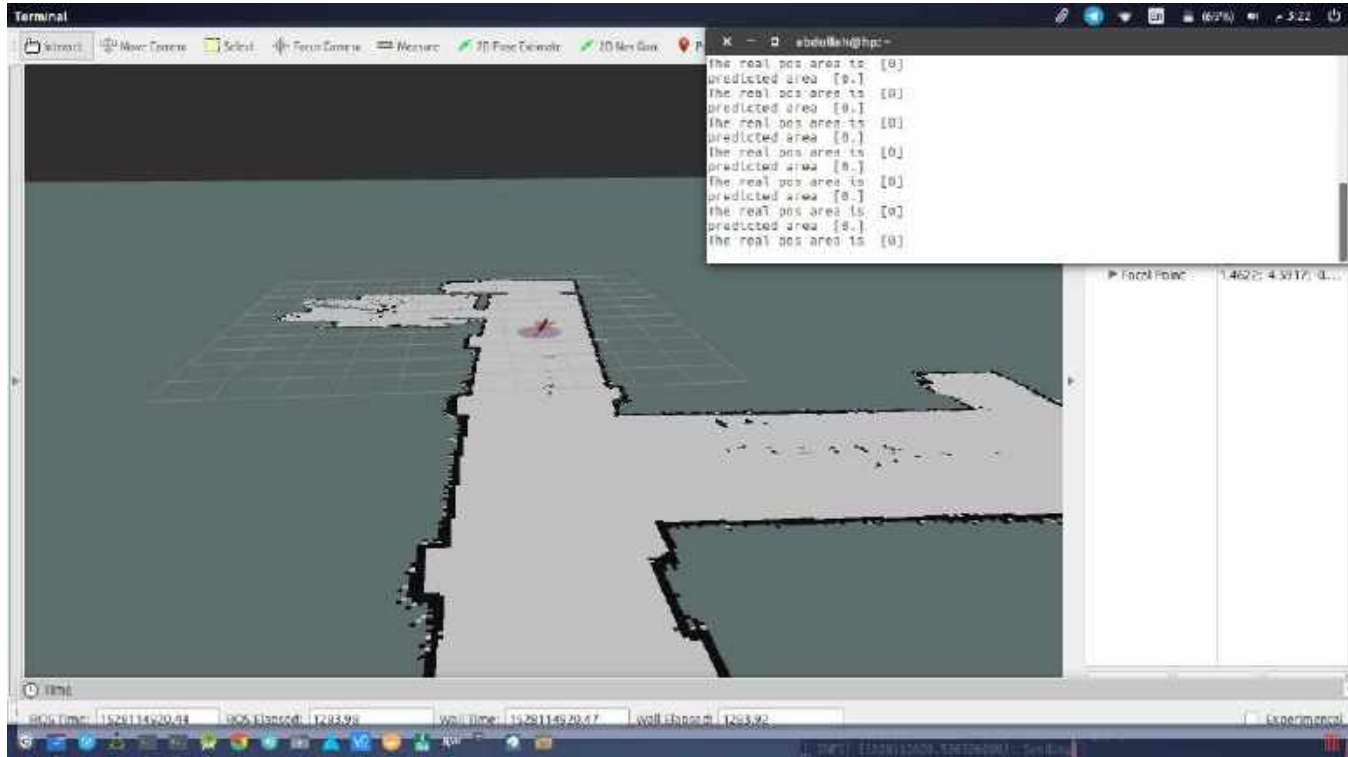




الشكل 8 — تقسيم الخريطة إلى عناقيد

والآن علينا وضع النظام ضمن حيز التنفيذ إذ سننشئ عقدة جديدة في نظام ROS اسمها area\_locator مسؤولة عن تشغيل النموذج عملياً وعند تحديد الموقع يجب أيضاً تشغيل عقدة Master وتطبيق GSM Data بشكل رئيسي ويمكن تشغيل الخريطة كما في الشكل 8-8 حيث

تم استخدام ثلاث مناطق: بداية الخريطة تمثل بالمنطقة 0 ووسطها بالمنطقة 2 وآخرها المنطقة 3



### الشكل 8 — 8 بيين خرج عقدة تحديد الموقع

والنظام حالياً يصنف الموقع حالياً بدقة حوالي 75 % تم حسابها يدوياً من عدة قياسات ولكن لحساب الخطأ بدقة يجب إنشاء عقدة في نظام ROS خصوصاً لتقديم تقرير الدقة Accuracy report يوجد فيه كل مقاييس التصنيف في جدول مناسب

## الخبرات المستفادة من المشروع

- بشكل رئيسي تم تطبيق خوارزميات تعلم الآلة عملياً على مشكلة تعدد صعوبة، وليس من السهل استخراج الأنماط ضمن الإشارات الراديوية اللاسلكية حيث أنها عرضة بشكل كبير للضجيج.
- إيجاد طريقة جديدة في نظم تحديد الموقع المعتمدة على تعلم آلة حيث تم إثبات قدرة تقنيات kernel approximation على إعطاء أفضل النتائج في المحاكاة وفي التطبيق على روبوت بشكل عملي.
- التعامل مع معطيات شركة حقيقية والعمل عليها.
- القدرة على بناء أنظمة تحديد الموقع ضمن المنشآت Indoor.
- كيفية بناء الخرائط والطرق الفعالة لفعل ذلك

## الآفاق المستقبلية

- المشروع أصبح يشكل قاعدة أساس لنظام تحديد الموقع والدقة المستحصلة هي دقة ابتدائية حيث يمكن زيادة الدقة بعدة طرق نذكر منها:
  1. زيادة عينات التدريب إذ يعد العدد المستحصل صغيراً في المشروعنا مقارنة بالأبحاث في هذا المجال وهذا ليس بالصعب إذ تتم العملية بشكل تلقائي كما ذكرنا سابقاً وإنما تحتاج لوقت ومدخلة روبوت جيدة
  2. إشارات الضجيج لا بد منها في هذه النظم ويمكن ببساطة إعادة تطبيق مرشح كالمان على النظام أو الأفضل تمثيل موقع الروبوت بطبقات ماركوف وهذا سيرفع وثوقية الأداء بشكل عالي.
  3. يمكن الاستفادة من إشارات Wi-Fi ذات المصادر الإشارة الثابتة حيث لا توجد في مبنى الحواسيب إشارات Wi-Fi بشكل يستفاد منه.
  4. تعديل عتادي على الشبكة بإضافة حساسات جديدة بشكل مناسب
- بناء الخريطة للمبنى بحيث تكون لعدة طوابق ويصبح النظام يشمل تحديد الإحداثيات والطابق الموجود به الروبوت.
- هذه الطريقة يمكن استثمارها بطرق سريعة جداً وعالية الدقة بحيث يتم تحديد الموقع ضمن العنقود أو المنطقة من شبكة الحساسات ومن ثم استخدام مرشحات ك Particle Filter لتحديد الموقع بدقة عالية ضمن هذا العنقود.
- إنشاء عقدة مسؤولة عن تقديم تقرير دقة عند كل اختبار.

## المراجع

1. Ali Rahimi, Benjamin Recht. 2007. "Random Features for Large-Scale Kernel Machines." Advances in Neural Information Processing Systems 20. Canada.
2. Dipanjan Sarkar, Raghav Bali, Tushar Sharma. 2018. Practical Machine. Apress.
3. Donges, Niklas. 2018. The Random Forest Algorithm. Towards Data Science. <https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>.
4. 2018. The Random Forest Algorithm. Towards Data Science. 2 22. <https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>.
5. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion. 2011. "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research 12.
6. Google. n.d. Android Developers. <https://developer.android.com/>.
7. Haykin, Simon. 2009. Neural Networks and Learning Machines. Pearson Education.
8. Musco, Christopher. 2018. Nonlinear dimensionality reduction for faster kernel methods in machine learning. Institute for Advanced Study. February. <https://video.ias.edu/csdm/2018/0212-ChristopherMusco>.
9. Ng, Andrew. "Support Vector Machines." In CS229 Machine Learning. Stanford.
10. ROBOTIS e-Manual. ROBOTIS. <http://emanual.robotis.com/>.

11. ROS. <http://www.ros.org/>.
12. Saleh, Yayha. 2017. Cellular-based Tracking System using Machine Learning Algorithms and Kalman Filter.
13. Sandy Mahfouz, Farah Mourad-Chehade, Paul Honeine, Joumana Farah, and Hichem Snoussi. 2014. Target Tracking Using Machine Learning and.
14. Simon, Dan. 2001. "Kalman Filtering, Embedded Systems Programming." 14.
15. Søren Riisgaard, Morten Rufus Blas. 2005. "SLAM for Dummies (A Tutorial Approach to Simultaneous Localization and Mapping)." 22.
16. Thampi, Sabu M. 2014. Advances in Intelligent Systems and Computing. Springer.
17. Trevino, Andrea. 2016. Introduction to K-means Clustering. oracle. 06 12. <https://www.datascience.com/blog/k-means-clustering>.

## **Abstract:**

In this project, we provide a new vision to benefit from the advances in machine learning algorithms that used in localization and tracking systems which depends on using wireless signals. The project includes a study to relate Received Signal Levels with the corresponding location and then getting the optimal estimate through Kalman Filter assuming that the user's motion and the measurement process are stochastic processes. Also, a practical outdoor implementation in Mazzeh Area, Damascus collaborating with telecommunication provider Syriatel and indoor implementation in Computer Engineering Building in Electrical and Mechanical Engineering faculty by using a mobile robot.

Damascus University

Faculty of Mechanical & Electrical Engineering

Department of Computer Engineering and Automation



# **Machine Learning Advances in Localization and Tracking Systems**

**Graduation Project prepared to obtain a Bachelor of Engineering Degree in Computer Engineering  
and Automation**

**By**

**Abdullah Al Saidi**

**Supervised by**

**Dr. Eng. HIYAM KHADDAM**

Academic Year

2017-2018