

**Teknik Rapor**

**Hazırlayan birim:** Kontrol Güdüm ve seyrüsefer Teknolojileri - MTKS

**Çalışma konusu:** Sabit Kanatlı İnsansız Hava Aracının uçuş esnasındaki temel bilgileri ve uyarıları içeren algoritmanın tasarlanması ve görselleştirilmesi

**Raporu Hazırlayan:** Yakup Gündüz

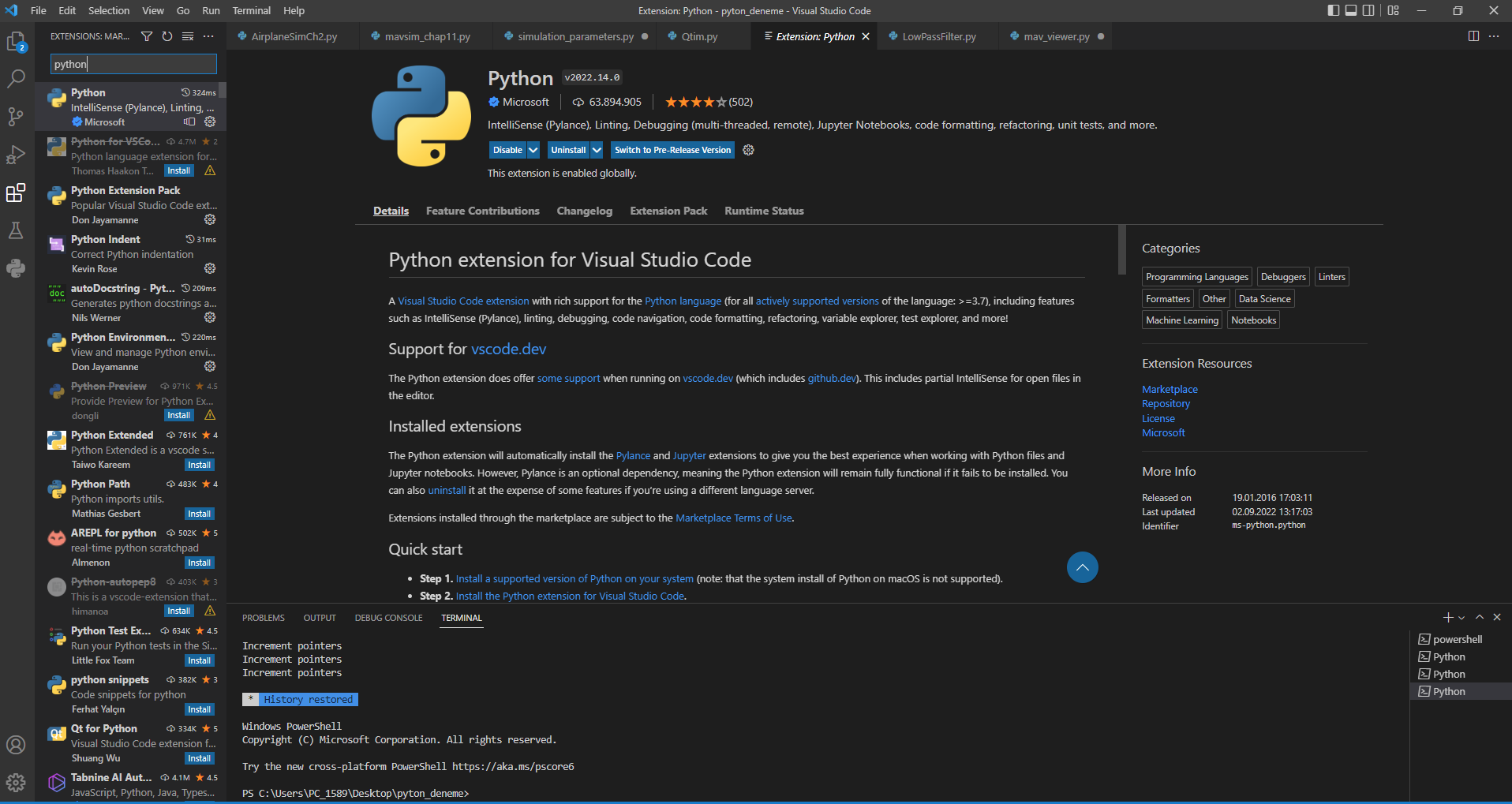
**Rapor Teslim Tarihi:** 16.09.2022

1. **Giriş**

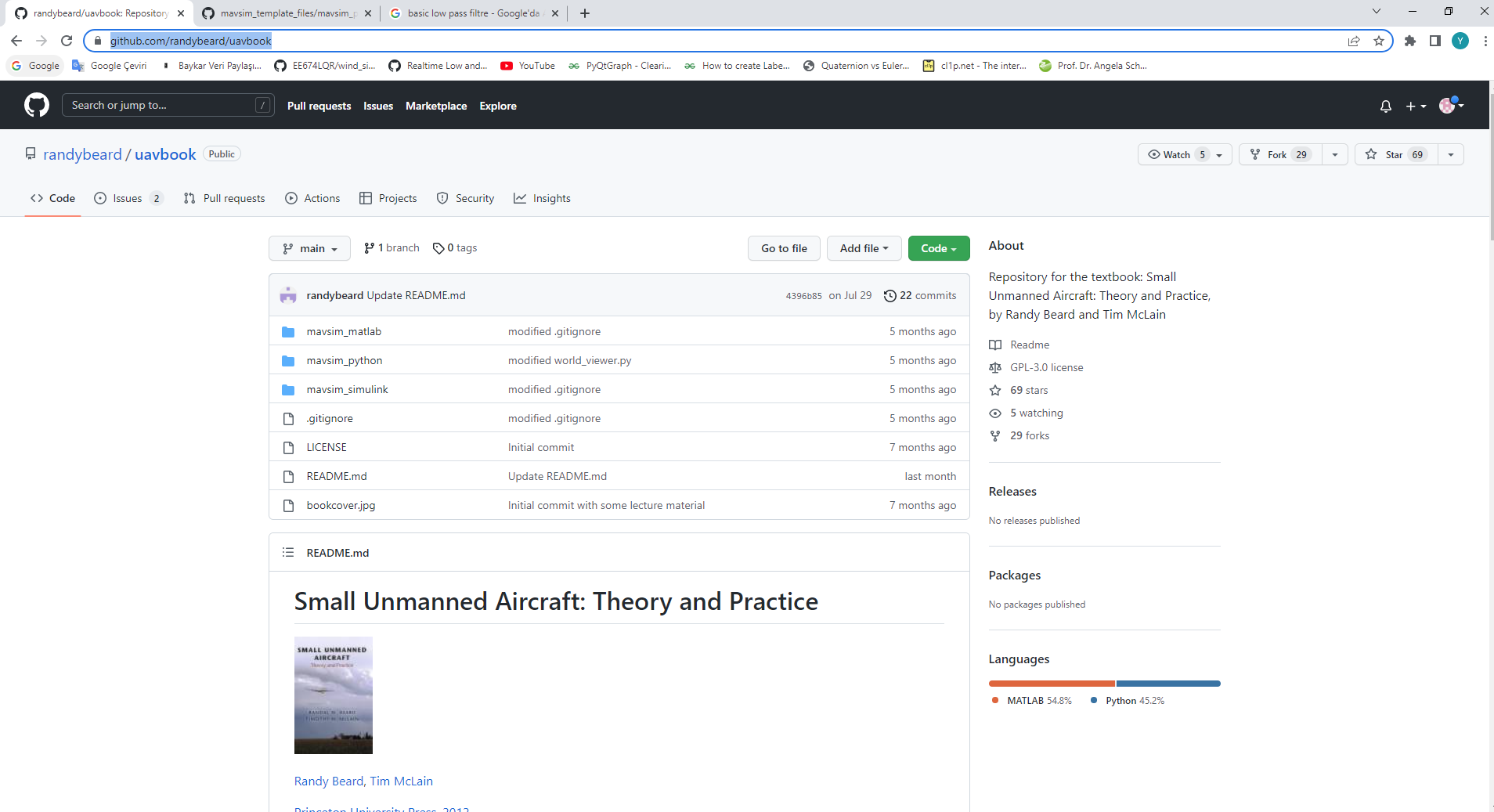
Yapılan çalışmada kullanılan model için Small Unmanned Aircraft kitabından yararlanılmıştır. Bu kitaptan yararlanılmasındaki temel sebeplerden biri github üzerinden çalışmaları içeren bir şablon bulunmasıdır. Bu kitaptan yararlanarak şablona uygun bir şekilde doldurulmuştur. şablona erişmek için buradaki [bağlantıyı](https://github.com/randybeard/uavbook) kullanabilirsin. bu linkte Python, Matlab ve simulink olmak üzere 3 tane şablon bulunmaktadır. Yapılan çalışma için Python tercih edilmiştir.

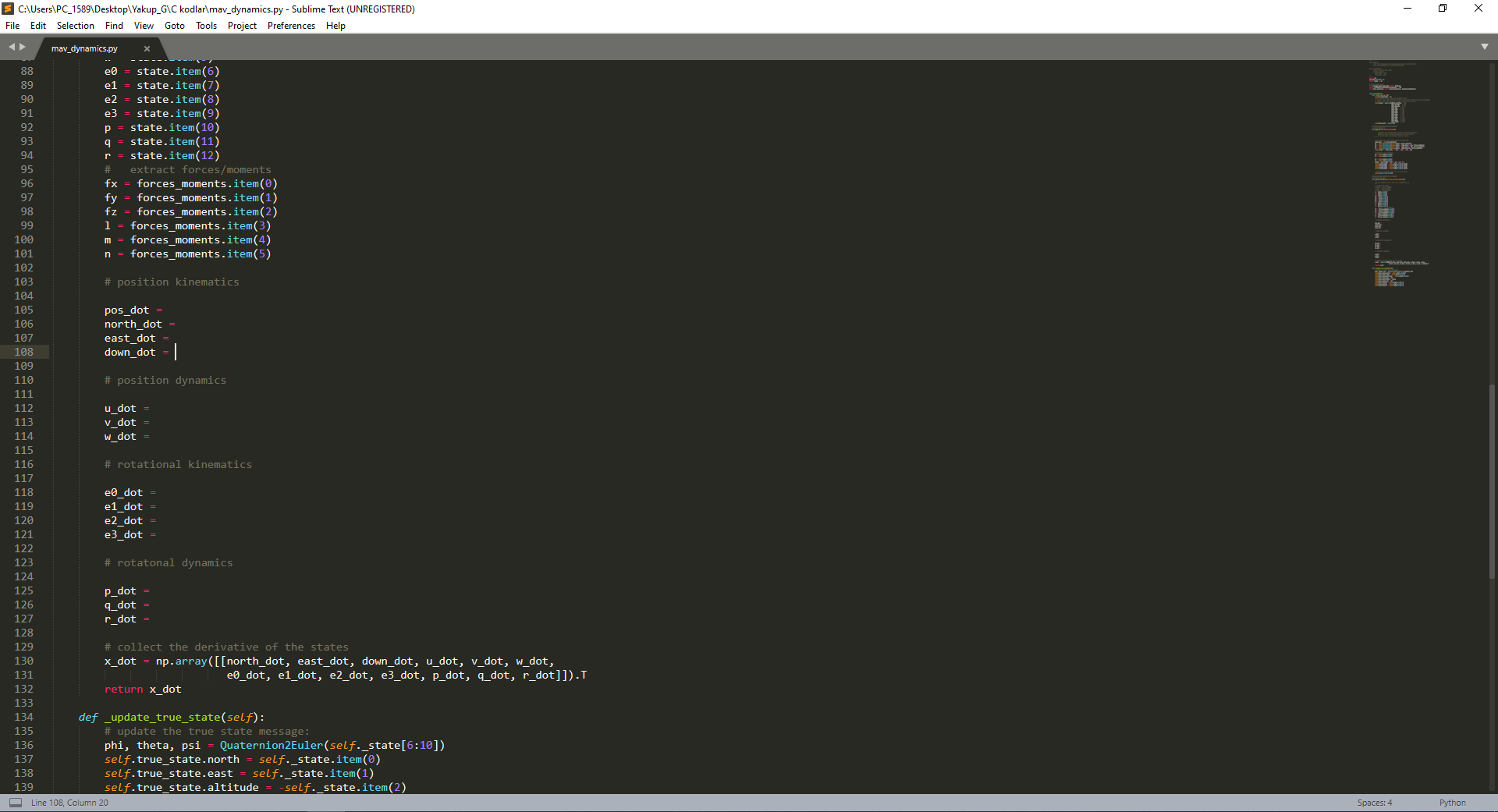
yapılan çalışma Windows 10 işletim sisteminde ve Visual Studio Code uygulaması üzerinden gerçekleştirilmiştir.

Visual studio programını kurmak için <https://code.visualstudio.com/download> adresinden windows 10 için uygun programı indiriniz daha sonra Visual Studio Code uygulamasını kurunuz üzerinden Python kurulumu gerçekleştirilmelidir. Python kurulumu için visual studio programını açtıktan sonra solda bulunan extension sekmesineden Python kurulumu için arama yapmanız gerekmektedir. ekran şekil X‘de gösterilmektedir. Daha sonra buradan Python için install tuşuna basarak kurulmasını bekleyiniz. şekil X‘deki resimde kurulduktan sonraki gösterilmektedir. Buna ek olarak yazım kolaylığı sağlamak için python extension pack’de kurabilirsiniz



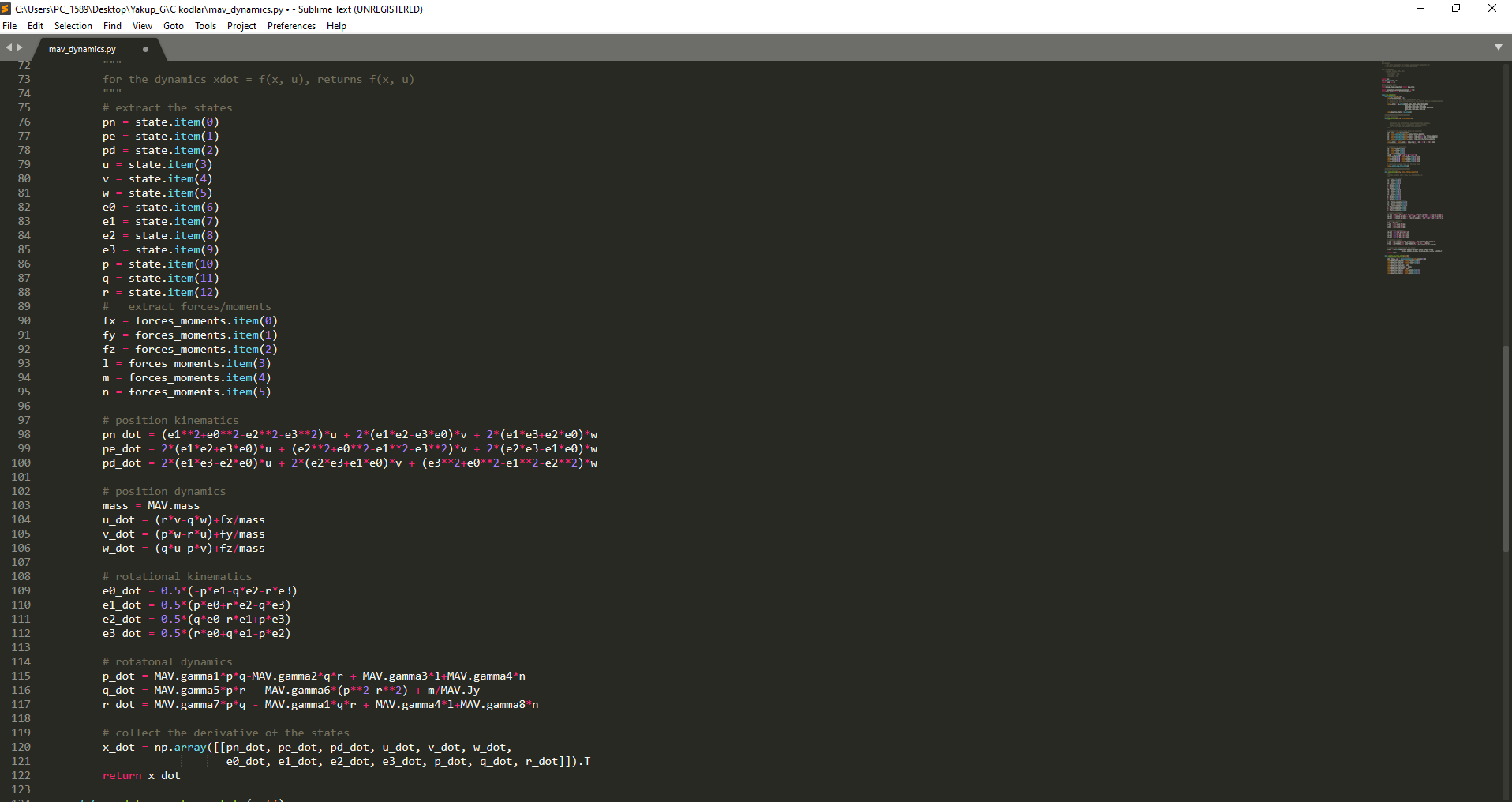
bir sonraki aşama olarak verilen github linkinden Python dosyası indirilmesidir. Burada öncelikle sağ üste bulunan code bölümünden tıklayarak dosyaların indirilmesidir



kitaptaki chapterlara bakılarak gerekli alanlar doldurulmalıdır. Normalde şablonların içlerinde belirli bölgeler boştur kitaptaki her bir chapter bu bölgeleri doldurmak için dökümandan yararlanmanızı beklemektedir. Şekil X’de mav\_dynamics dosyasında Position Kinematics, Position Dynamics, Rotational Kinematics ve Roratonal Dynamics alanları uygun şekilde doldurulmalıdır.

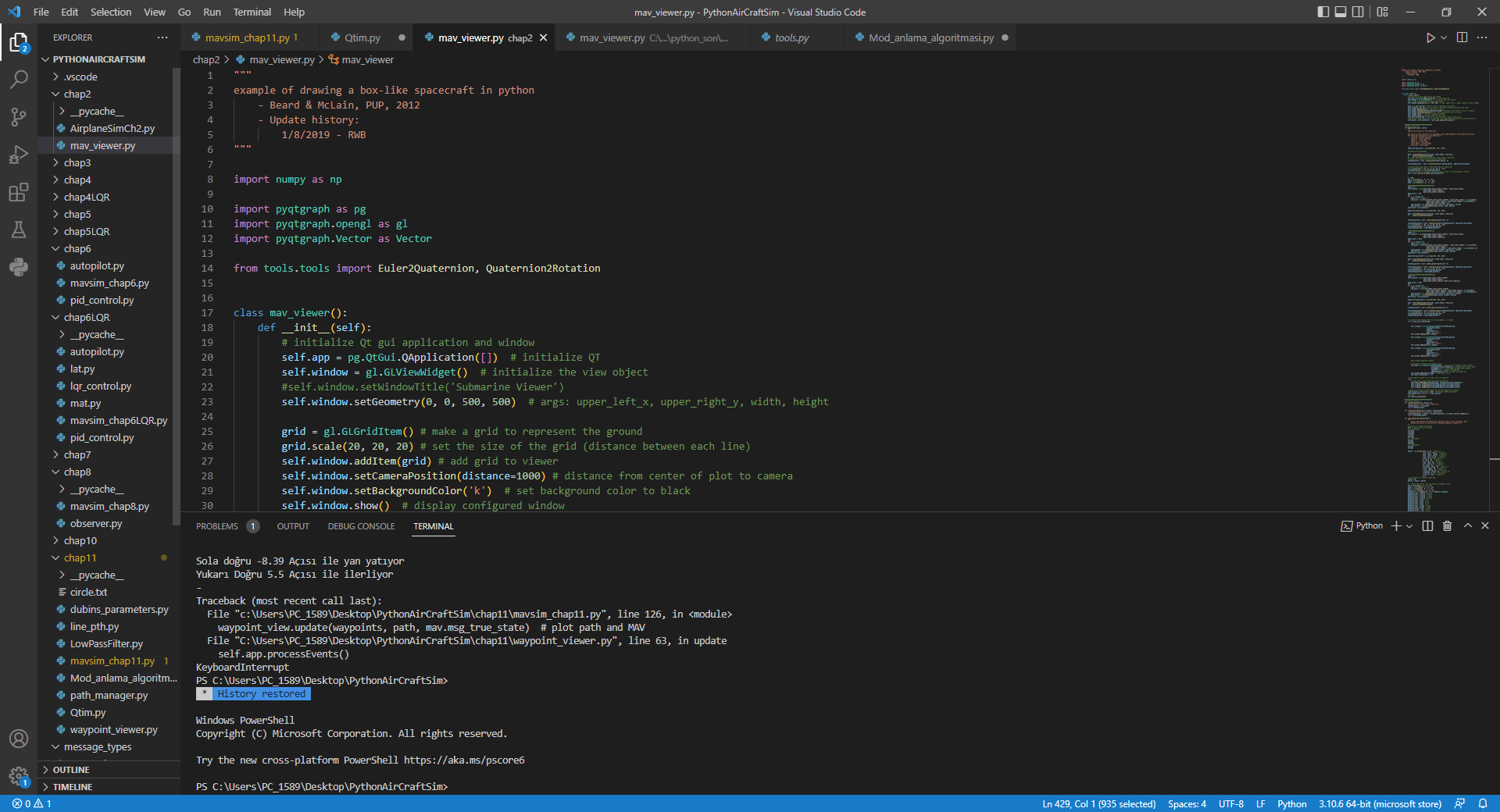
kitapta bu bölüm chapter 3 sonunda equation of motion summary bölümünde gösterilmektedir. Buradaki hesaplamalar verilen şablondaki değişkenlere göre yazılmalıdır.

Verilen Python kod dosyasını buna göre düzenlediğmizde aşağıdaki gibi olmaktadır.



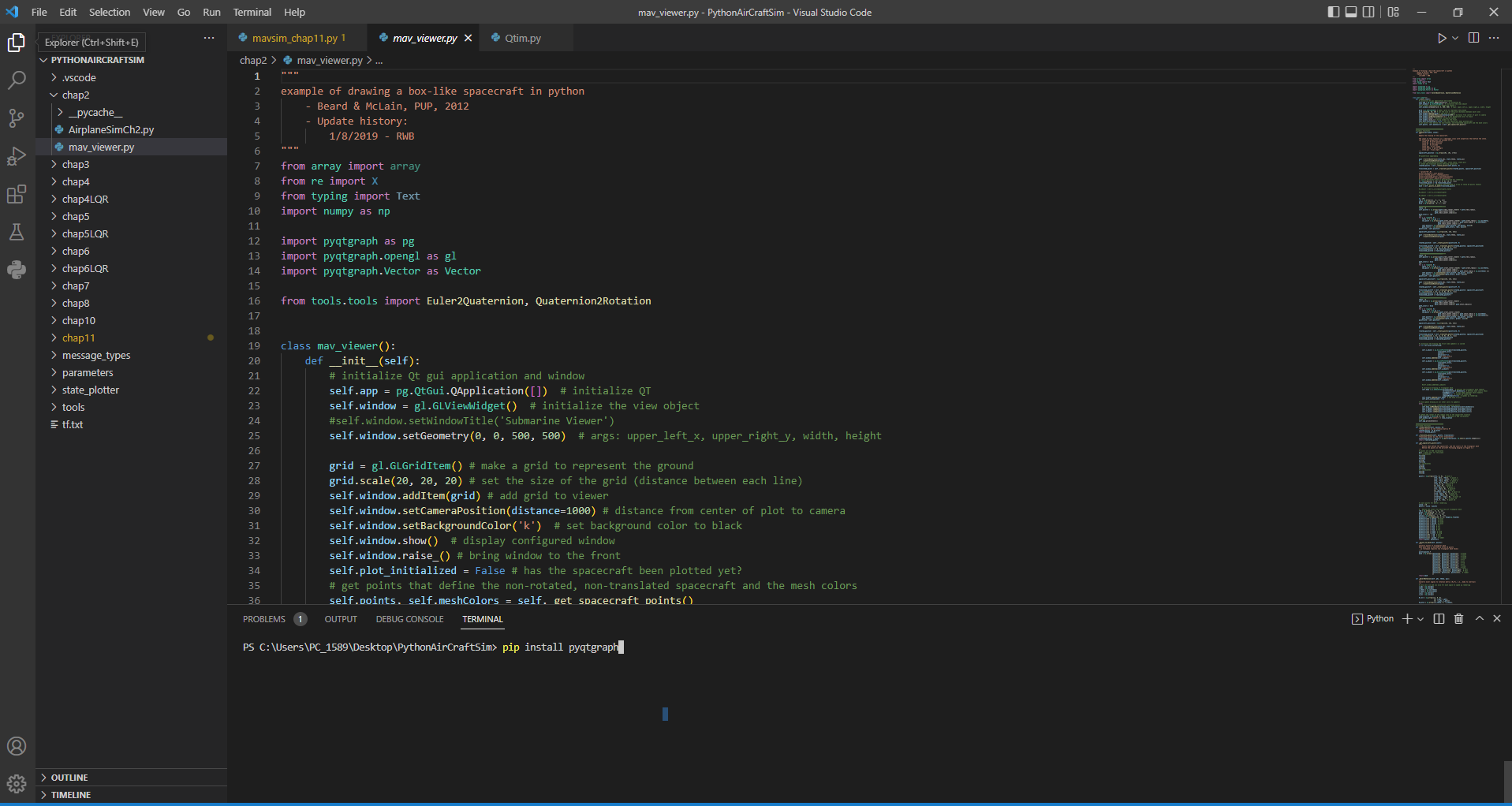
Yapılan çalışmada bu işlemler yapılmış olup tekrardan yapılmasına ihtiyaç yoktur. Hepsi hazır bir şekilde dosya haline getirilmiştir. eğer yapılan çalışmaya ulaşılamıyorsa bu yönteme başvurulabilir.

Bu işlemler yapıldıktan sonra kaynak dosya çalıştırırken bazı kütüphaneler kurulmalıdır bu kütüphaneler kaynak dosyalarda en üste gösterilmektedir. Kodu derlemeye çalıştığınızda size eksik kütüphaneleri göstermektedir. Buradaki eksik kütüphaneleri bu şekilde belirleyebilirsiniz



Şekil x : üstte gösterilen kütüphanelerden bir tanesi

Kütüphanelerin kurulma işlemi terminale “pip install pyqtgraph” yazarak yapılmaktadır “pyqtgraph yerine hangi kütüphaneyi kurmak istediğinizi belirtmelisiniz



Kütüphanelerin kurulumu tamamlandıktan sonra sistemdeki dosyaların birbirini algılayabilmesi için

Sys.path.append komutunu kullanmalısınız bu komut ile farklı konumlardaki dosyaların birlikte çalışmasını sağlayabilirsiniz şuanda kullanılan bilgisayarda bütün dosyalar Desktop\PythonAirCraftSim içinde olduğu için bu şekilde yazılmıştır kendi dosyanızın bağlantı yoluna göre ayarlanmalıdır. Bazı bilgisayarlarda append parantezi içindeki r komutu kullanılmadığı görülmüştür. Eğer sistem çalışmamış ise r parametresini silebilirsiniz yine aynı şekilde sonuçlanırsa bu konuyu araştırabilirsiniz.



Kütüphane kurulumları ve bağlantı noktaları eklendikten sonra simülasyon ortamı çalışmaya hazır durumdadır.

Yapılan algoritma arayüz ve arayüz olmadan kullanım için iki tane kütüphane yazılmıştır bunlar Mod\_anlama\_algoritması.py ve Qtim.py dosyalarıdır Qtim arayüz içeren algoritmadır.

Algoritma 5 farklı bilgi içermektedir. Bunlar düz gitme, daire çizme, irtifa kontrol, yön takip ve hareket kontrol bilgileridir bu algoritmaların hava aracının hareket bilgileri algılamada yeterli olduğu görülmüştür.

**Algoritmalar**

1. **Düz gitme**

bir hava aracının düz gittiği anlamanın en basit yolu heading açısının değişimine bakmaktır.

Eğer heading acısı değişiyorsa araç düz rotadan çıkmaktadır.

Heading offsett değeri heading önceki değerinden belirbir değerden büyük ise düz rotadan cıktığı kabul edilmektedir.

1. **Daire çizme**

Bir hava aracının çizdiği yarıçapı belirlemek için aracın hızını ve pitch açısını bilmek gereklidir bu değerleri bildiğimizde eşitlik (x)’den yararlanarak ne kadarlık bir yarıçap’da daire çizdiğini hesaplayabiliriz.

(x)

Buradaki R yarıçap formülü direk olarak aracın çizdiği yarıçapı göstermektedir. Bir rotada düz giderken bile aracın sağa veya sola doğru yönelimleri yarıçap oluşturduğunu düşünmeye yol açabilir bunu engellemek için aracın küçük yönelimlerini algılamamak için belirli bir offsett değeri belirlemek gerekmektedir. bu offsett sayesinde aracın yaptığı açı değeri ofsett değerinden büyük olduğunda bir dönüş yapmış kabul edilir.

1. **İrtifa kontrol**

Bunun öncelikle verideki gürültüyü sönümlemek amacıyla alçak geçiren filtre kullanılmıştır detayları 3.1’de anlatılmaktadır. Alçak geçiren filtrenin uygulandığı altitude değeri aşağıdaki formül ile değerinin üstündeyse yükseklikte gözle görülür bir değişim var demektir ve buda bir çıkarım yapmamız gerektiğini göstermektedir aşağıdaki eşitlikte gösterilmektedir.

(x)

Bu koşul sağlandığında

eğer ise irtifa kazanıyor diyebiliriz

eğer ise irtifa kaybediyor diyebiliriz

h değeri yükseklik bilgisini temsil etmektedir.

* 1. **Alçak geçiren filtre**

Alçak geçiren filtre (LPF), seçilen bir kesme frekansından daha düşük bir frekansa sahip sinyalleri ileten ve kesme frekansından daha yüksek frekanslı sinyalleri zayıflatan bir filtredir.[1] eşitlik x’de birinci derece bir alçak geçiren filtre denklemi verilmektedir.

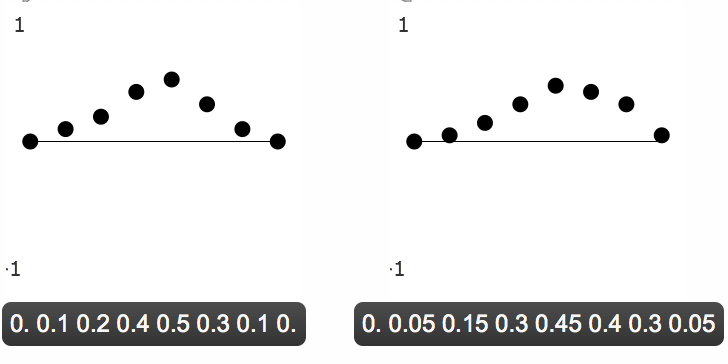
(x)

Burada 0.1 değeri alçak geçiren fitrenin frekansına bağlı olarak alınmalıdır. Temel mantık yeni hesaplanan değeri anında kabul etmeyip bunu zamana yamaktır diyebiliriz**.** Örneğin gürültü bir veri var frekans arttıkça değeri daha hızlı bir şekilde istenen değeri yakalayacak ama frekans azaldıkça hesaplanan değere ulaşmakda bir o kadar zorlaşacak ve verinin istenen değere yaklaşması zorlaştığından aslında veriyi geçirmemiş olarak kabul edebiliriz. Burada a değerini düşürdükçe alçak geçiren frekansı yükselmektedir. Aralarında aşağıdaki gibi bir oran bulunmaktadır.

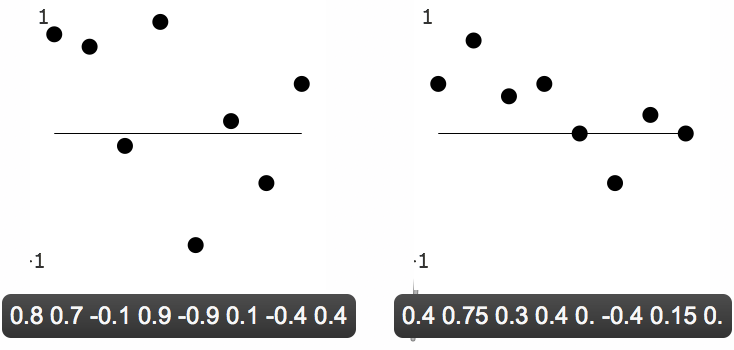
(x)

Alçak geçiren frekans hesabı = 1 / (1-a) dan çıkarılmaktadır. Grafik olarak göstermek gerekirse aşağıdaki grafikte görüldüğü üzere a değerini bağlı olarak istenilen frekans filtrelenebilir. Örneğin 10 hz bir sinyal filtrelenmek isteniyor a değeri 0.9 seçilerek 10 hz lik bir sinyal sönümlenebilir. (bunun hesabı doğrulanmadı)

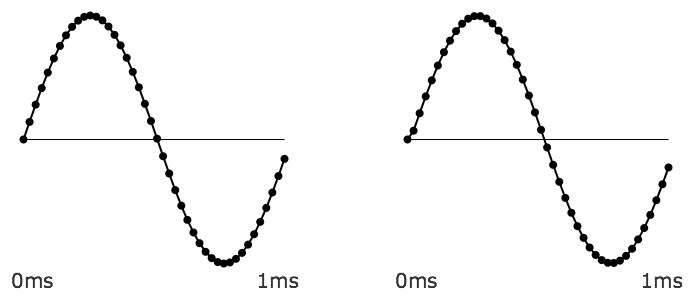
Bu grafik 0.01 aralıklarla ifade edildiği için 100 hz’ e kadar filtreliyor gibi gözükmektedir semboliktir ve gerçeği yansıtmamaktadır. Sadece a değeri ve frekans ilişkisin göstermek amacıyla yapılmıştır



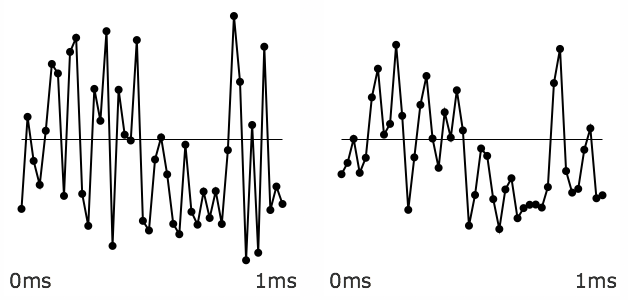
Şekil X:Pürüzsüz bir giriş, alçak geçiren filtre tarafından yalnızca hafifçe düzleştirilir



Şekil X: Pürüzlü bir giriş, alçak geçiren filtre tarafından daha belirgin şekilde yumuşatılır



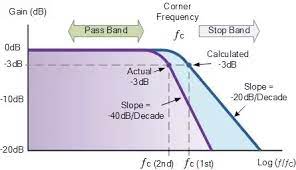
Şekil X : Alçak geçiren filtreden neredeyse tamamen etkilenmeyen 1KHz sinüs tonu



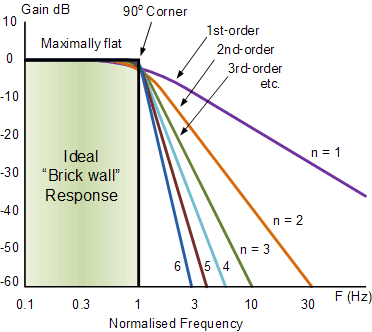
Şekil x: Alçak geçiren filtre tarafından gürültülü bir giriş daha fazla değiştirilir*.*

Yukarıda anlatılan filtre birinci derece alçak geçiren bir filtredir. Eğer formüldeki + ifadesini – yapıldığında yüksek geçiren bir filtre elde etmiş olursunuz

Filtrelerin dereceleri vardır ve bu dereceler arttıkça filtrenin -dB cinsinden gücüde artmaktadır. Aşağıdaki grafik filtrenin derecesi ve gücünü göstermektedir.



-3dB olayı önemli





Görüldüğü üzere filtrenin derecesi arttıkça kesim frekansında daha doğru sonuçlar vermektedir. fakat Filtrelerin derecesinin artması bazı kötü özellikleri vardır. Bunlar sırasıyla verinin amplitüde düşmektedir ve zamanda geriye kaymaya başlamaktadır bunlarda istenemeyen durumlar ortaya çıkarmaktadır. Sinyalin genliğinin azalması sinyalin bozulmasına sebep olurken zamanda kaymasıda veriden oluşturacak çıktının gecikmesine sebep olmaktadır. Bu yüzden filtrelenecek veriye göre bu dereceler ayarlanmalıdır. Yapılan araştırmalarda havacılıkda genelde 2. Derece filtre kullanıldığı gözlemlenmiştir.

ikinci derece low pass filtre yapmak isterseniz aşağıdaki linklerden yararlanabilirsiniz (detaylandırılabilir!!!)

<https://doctorpapadopoulos.com/butterworth-low-pass-filter-c-c-implementation/>

<https://stackoverflow.com/questions/664877/i-need-to-implement-a-butterworth-filter-in-c-is-it-easier-get-a-library-with-t>

<https://stackoverflow.com/questions/20924868/calculate-coefficients-of-2nd-order-butterworth-low-pass-filter/20932062>

**Yön takip:**

Yön takip algoritmasını kuzey kutbunu 0 kabul ettiğimizde güney kutbu 180 derecede kalmaktadır.

Buna bağlı olarak doğu, batı, güney ve kuzey yönlerinden hangisine hangi açıyla gittiğimizi gözlemleyebiliriz. Heading parametresini istenen sınırlada yapıldığında rahatlıkla çıkmaktadır.

**Not:** simülasyondaki algoritma 0-360 arasına scallenmemiştir -180 ile 180 arasındadır algoritma değerleri buna göre ayarlanmıştır.

**Hareket Kontrol:**

Hareket kontrolü 2 eksen üzerindeki hareketi algılayarak aracın yatış açıları hakkında bilgi edinmemizi sağlamaktadır. Bu sayede sabit irtifada giderken bile pitch vb hareketleri gözlemleyebiliriz. Öncelikle Roll ve pitch açısı için bir offsett değeri belirlenmelidir bunun sebebi roll ve pitch’deki ufak hareketlenmeleri gözardı etmektir.

Roll merkezi 0 kabul edilmiştir.

Üstteki denklem doğrulandığında

Eğer ise sağa doğru yatıyor değilse sola doğru yatıyor diyebiliriz

Aynı şekilde pitch içinde düşünelim

pitch merkezi 0 kabul edilmiştir.

Üstteki denklem doğrulandığında

Eğer ise yukarı doğru yatıyor değilse aşağı doğru yatıyor diyebiliriz

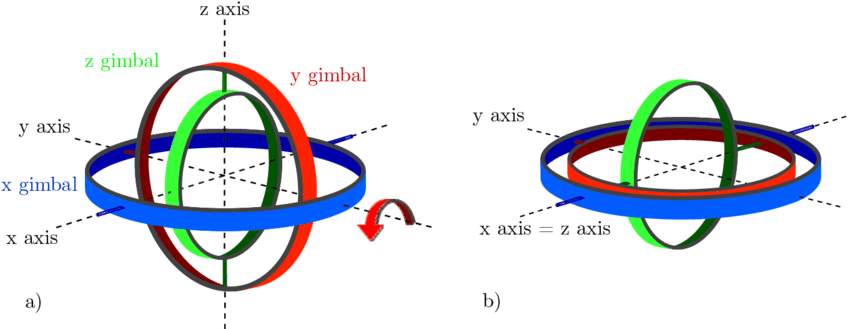
Rotasyon nedir ?

Rotasyon kısa bir tabirle döndürme eylemei olarak tanımlanabilir.

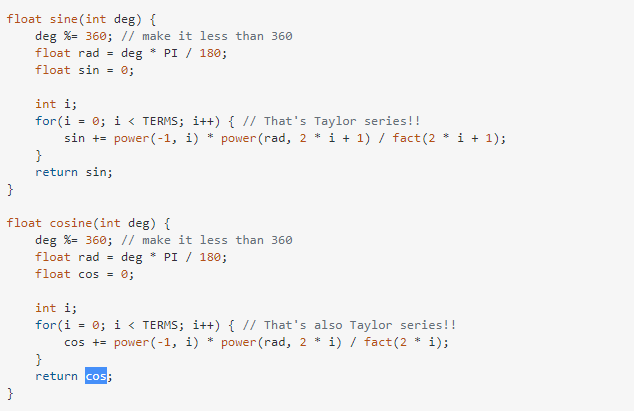
Cisimlerin merkezi etrafında yaptığı hareketlerin hepsine kapsamaktadır. Özllikle oyunlarda ve havacılıkta 3 boyutlu ortamda bir cismin dönüşünü ifade etmek için kullanılır. 2 farklı şekilde rotasyon yapılabilmektedir. bunlar euler ve quanternion methodlarıdır bu methodlar ve uygulanma şekilleri verilen dosyanın chap2 mav\_viewer.py dosyasında bulunmaktadır. şimdi bu iki methodun kullanımı avantajları ve dezavantajlarından bahsedelim euler açıları insan beyninin daha rahat anlayabileceği ve yorumlayabilmesi daha kolay bir methoddur ama içindeki matris ve cosinüs işlemleri bir bilgisayar ve makine için daha zordur aynı zamanda euler açısının bir sıkıntısıda gimbal lock etkisidir.

Apollo 11-13 görevinde bu sorun yaşanmış.

**Gimbal lock :** Üç gimbalin ikisinin eksenleri paralel bir konfigürasyona sürüldüğünde meydana gelen üç boyutlu bir uzayda bir serbestlik derecesi kaybı , sistemi dejenere iki boyutlu bir uzayda dönmeye "kilitliyor" .



Peki neden euler insanların anlaması bu kadar kolay iken neden quanternion tercih ederiz yada kullanırız. Asıl sorun bilgisayara göre quaternionun işlem maliyeti euler göre daha azdır bunun sebeplerine gelicek olursak eğer

Cosinüs ve matris işlemleri buna sebep olmaktadır. Aynı zamanda 

Matris işlemlerinin zorluğunu deneyimledim diyebilirim yada şey diyebilirsin matris işlemlerinde istenilen matrise erişmek için sürekli istenilen indise erişmenin cok fazla zaman harcadığından olabilir diye düşündüm internetde bunu üzerine direk bir makale bulamadım

Kaynakça:

[1]https://tr.wikipedia.org/wiki/Al%C3%A7ak\_ge%C3%A7iren\_filtre