

**Teknik Rapor**

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

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

**Raporu Hazırlayan:** Yakup Gündüz

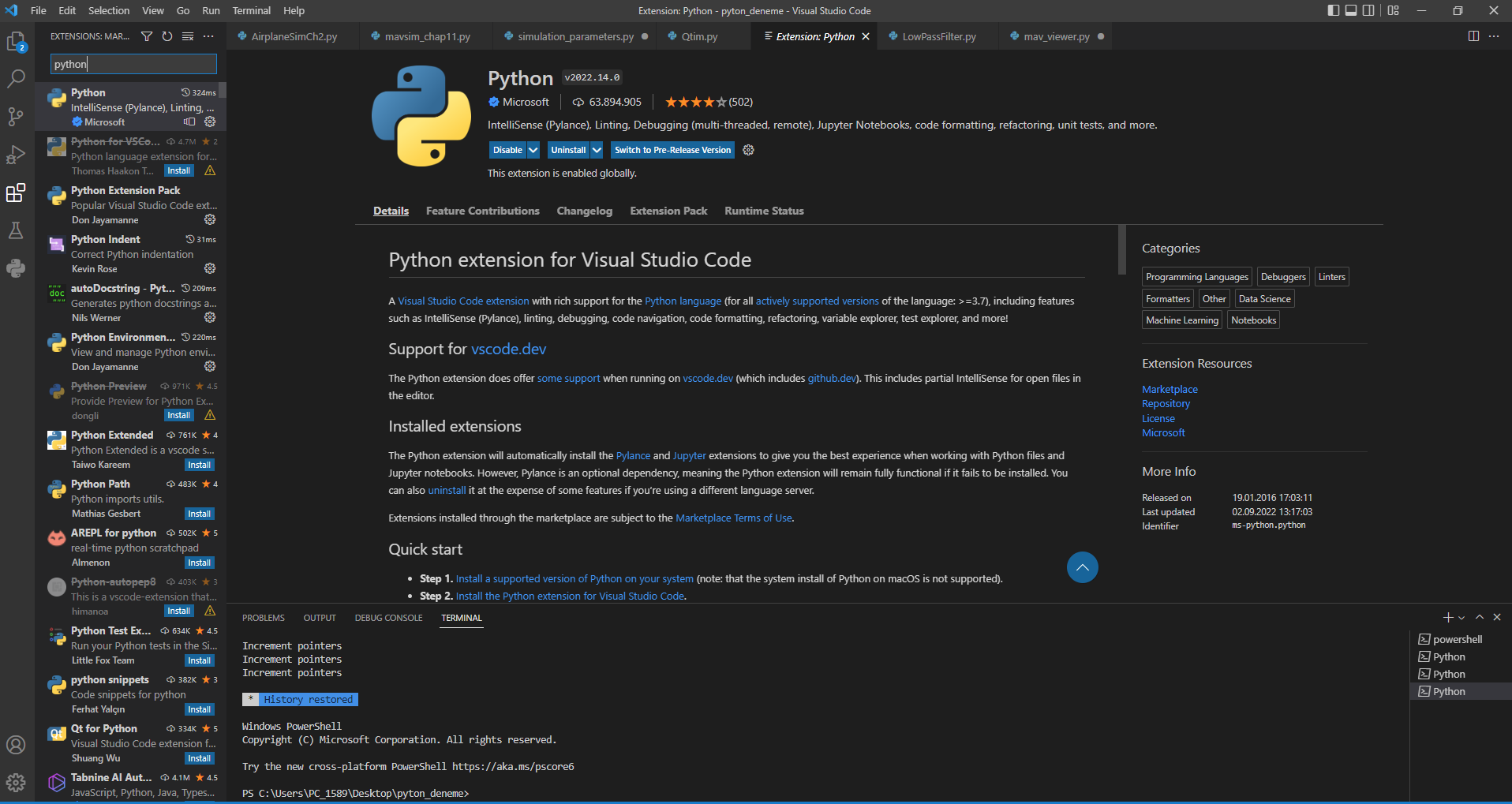
**Rapor Teslim Tarihi:** 16.09.2022

1. **Giriş**

Yapılan çalışmada kullanılan model için Small Unmanned Aircraft kitabından yararlanılmıştır. Bu kitaptan yararlanılmasındaki temel sebeplerden biri github üzerinden çalışmaları içeren bir açık kaynak kod taslağı bulunmasıdır. Bu kitaptan yararlanarak kod taslağı uygun bir şekilde doldurulmuştur. Kod taslağı erişmek için buradaki [bağlantıyı](https://github.com/randybeard/uavbook) kullanabilirsin. Bu linkte Python, Matlab ve Simulink olmak üzere 3 tane kod taslağı bulunmaktadır. Yapılan çalışma Python dili kullanılarak. İşletim sistemi olarak Windows 10 ve derleyici olarak Visual Studio Code uygulaması üzerinden gerçekleştirilmiştir.

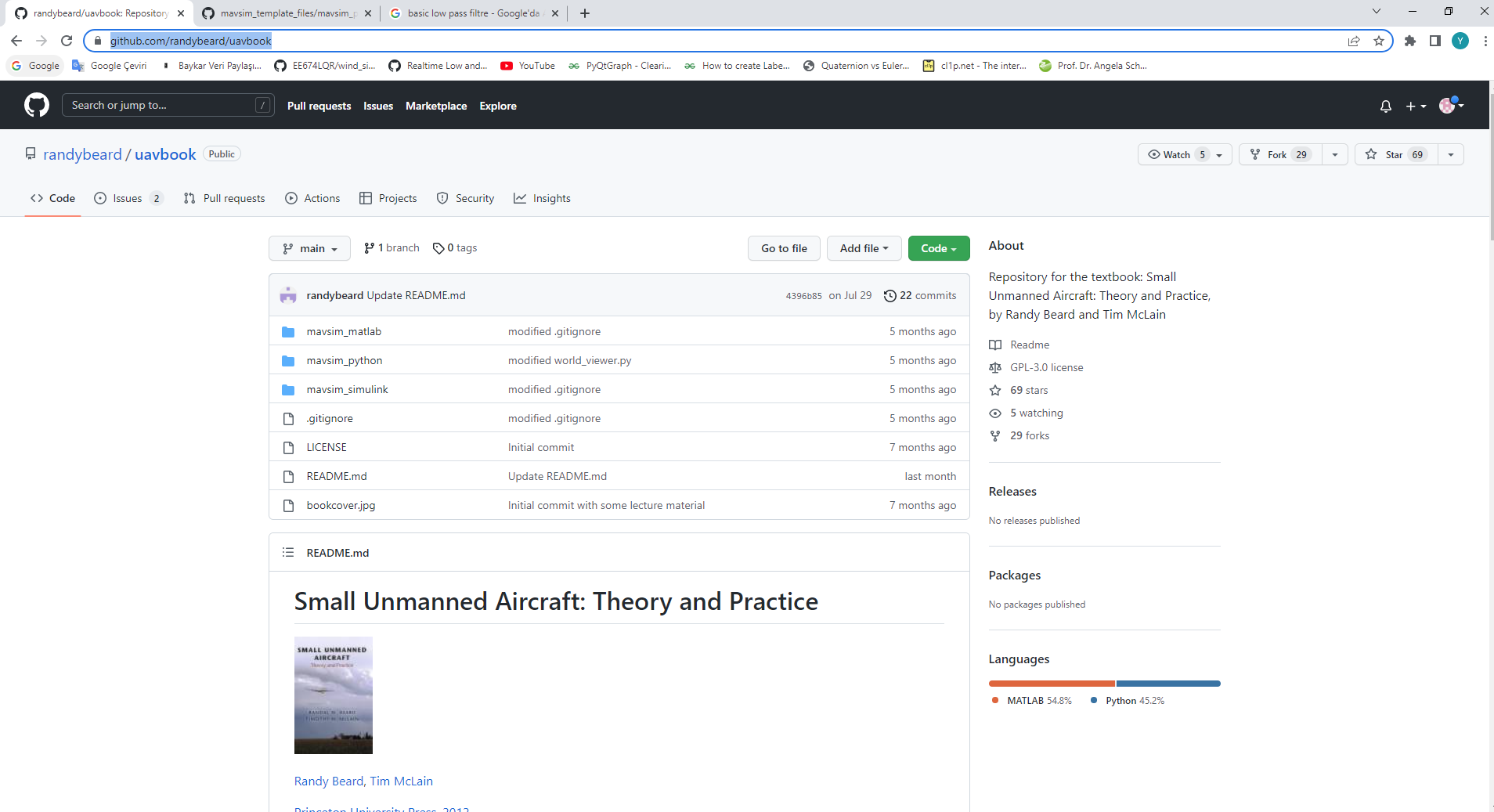
1. **Visual Studio Code ve Dosyaların Bağlantısının Yapılması**

Visual Studio Code programını kurmak için [bağlantıyı](https://code.visualstudio.com/download) kullanabilirsiniz. Bağlantı adresinden windows 10 ya da kendi işletim sisteminiz içim en uygun olanını indiriniz ve daha sonra Visual Studio Code uygulamasını kurunuz. Visual Studio Code uygulaması içinden Python kurulumu gerçekleştirilmelidir. Python kurulumu için Visual Studio Code programını açtıktan Şekil 2.1’de gösterilen sol üsteki sarı ile işaretlenmiş alan üzerinden Extension sekmesi ile Python kurulumu için arama yapmanız gerekmektedir. Daha sonra buradan Python için İnstall tuşuna basarak kurulmasını bekleyiniz. Şekil 2.1’deki resimde kurulduktan sonraki hali gösterilmektedir. Buna ek olarak yazım kolaylığı sağlamak için python extension pack’de kurabilirsiniz.



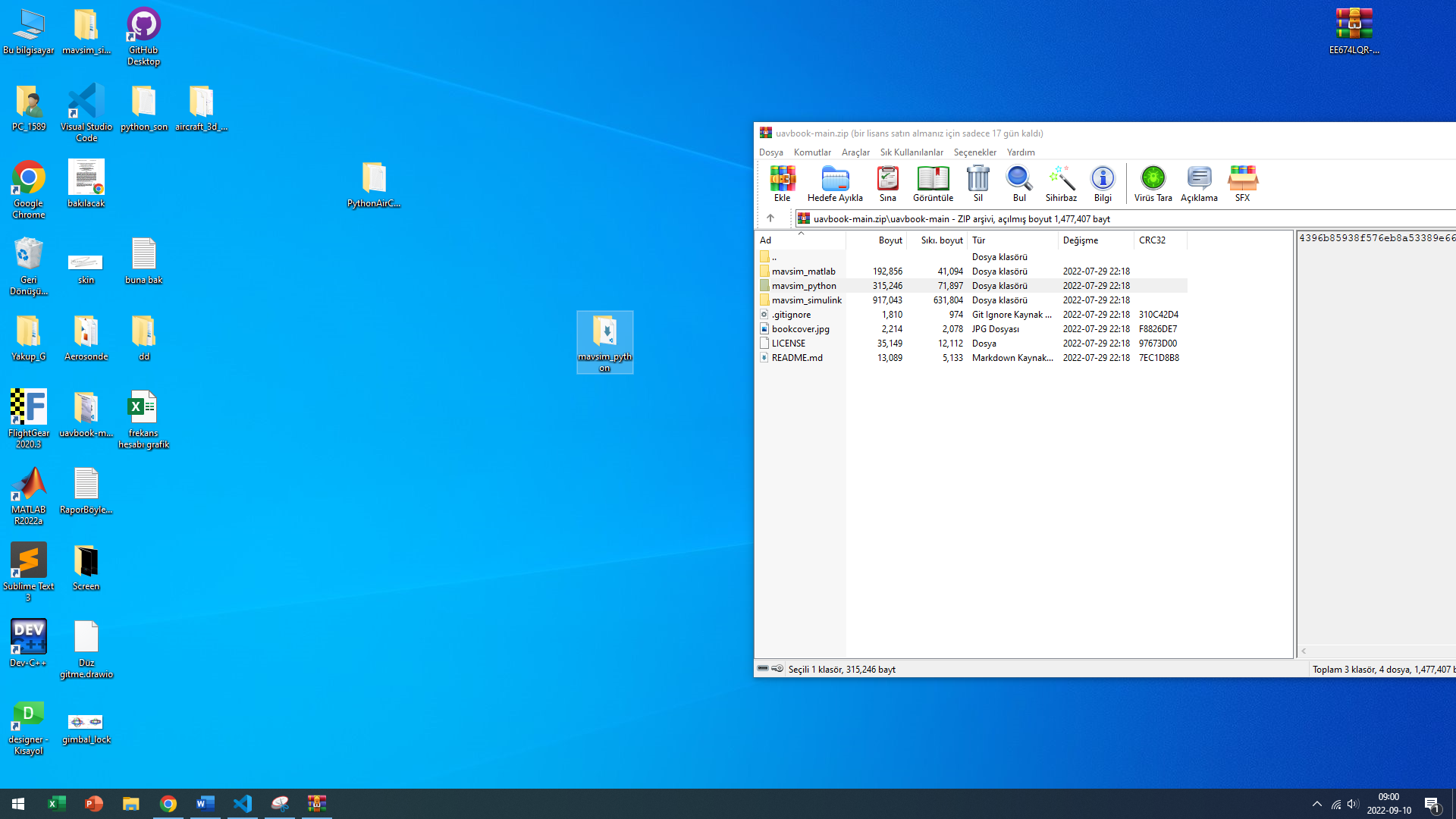
Şekil 2.1: Extension Sekmesi

Bir sonraki aşama olarak verilen github [bağlantısından](https://github.com/randybeard/uavbook) Python dosyası indirilmesidir. Burada öncelikle şekil 2.2’de sağ üste bulunan code bölümüne tıklayarak dosyalanın Zip şeklinde indirilmesidir.



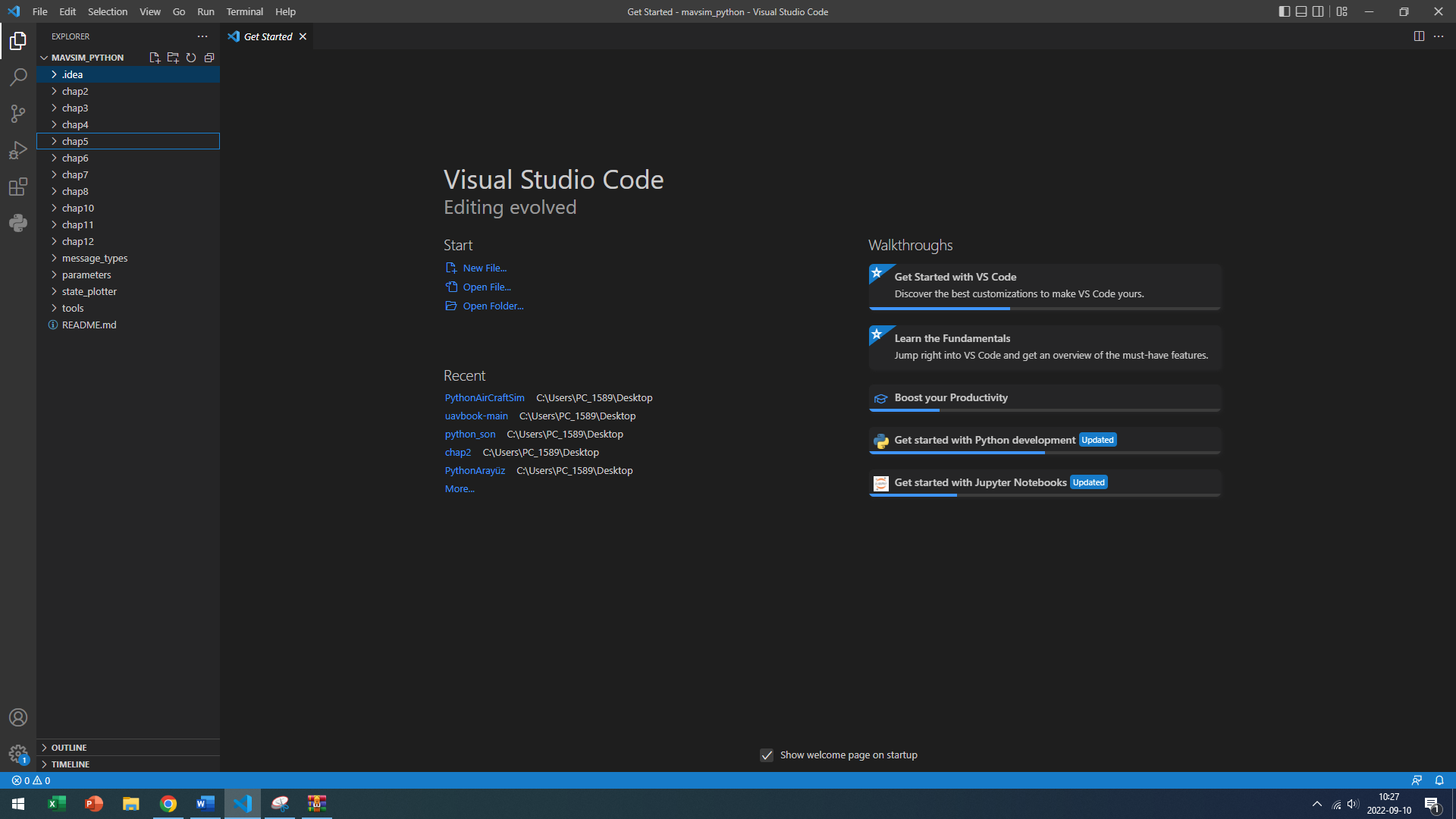
Şekil 2.2: Python Dosyası İndirilmesi

Zip halinde indirilen dosya açılarak “mavsim\_python” dosyasını istediğiniz bir konuma atabilirsiniz. Yapılan çalışma bu dosyayı masaüstüne atıldığı varsayılarak anlatılacaktır. Şekil 1.3’de masaüstüne atılan dosya gösterilmektedir.



Şekil 2.3: Masaüstüne Atılan “mavsim\_python” Dosyası

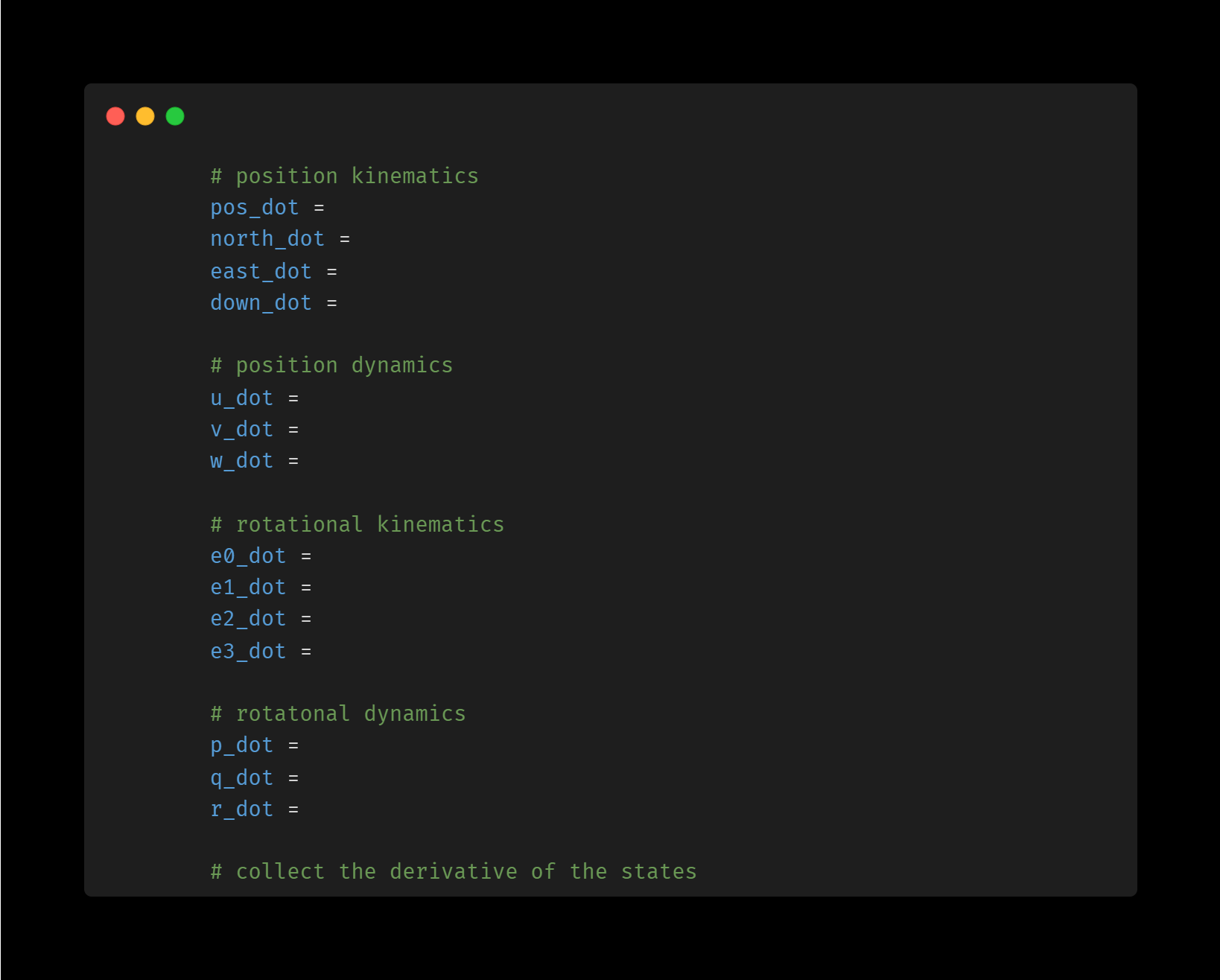
daha sonra masaüstündeki bu dosyanın bağlantısı Visual Studio Code üzerinden açılmalıdır. Bağlantının açılması için Visual Studio Code uygulaması içinden File->Open Folder şeçeneğine tıklayarak dosyanın konumunu seçiniz. Artık dosyanıza Visual Studio Code üzerinden erişebilirsiniz. Dosyanızı açtığınızda şekil 2.4’de sol üsteki sarı alandaki gibi gösterilecektir.



Şekil 2.4:Bağlantı Yolunun Açılmış Hali

1. **Unmmanned Small Aircraft Kitabına Uygun Şekilde Kod Düzenlenmesi**

Bu işlemlerden sonra Kod taslağındaki chapterların içinde şekil 2.4 ‘de görüldüğü üzere belirli bölgeler boştur. Kitaptaki chapterlar, Koddaki chapterların içlerini uygun şekilde doldurmak için matematiksel denklemler içermektedir. Bu bölgeleri doldurmak için kitapdan yararlanmanızı beklemektedir. Kitaptaki chapterlara bakılarak gerekli alanlar doldurulmalıdır.(Not: kitaptaki chapterlar verilen linktede chapterlar halinde bulunmaktadır). Bu chapterların doldurulması için örneğin Şekil 3.1’de mav\_dynamics dosyasında Position Kinematics, Position Dynamics, Rotational Kinematics ve Rotational Dynamics alanları uygun şekilde doldurulmalıdır.



Şekil 3.1:Chapter 3 mav\_dynamics’deki Boş Alan

Bu bölümde kitapta chapter 3 sonunda equation of motion summary bölümünde 4 tane eşitlik gösterilmektedir aşağıda 1,2,3 ve 4 numaralı eşitlik ile ifade edilmektedir. Buradaki hesaplamalar verilen şablondaki değişkenlere göre yazılmalıdır.

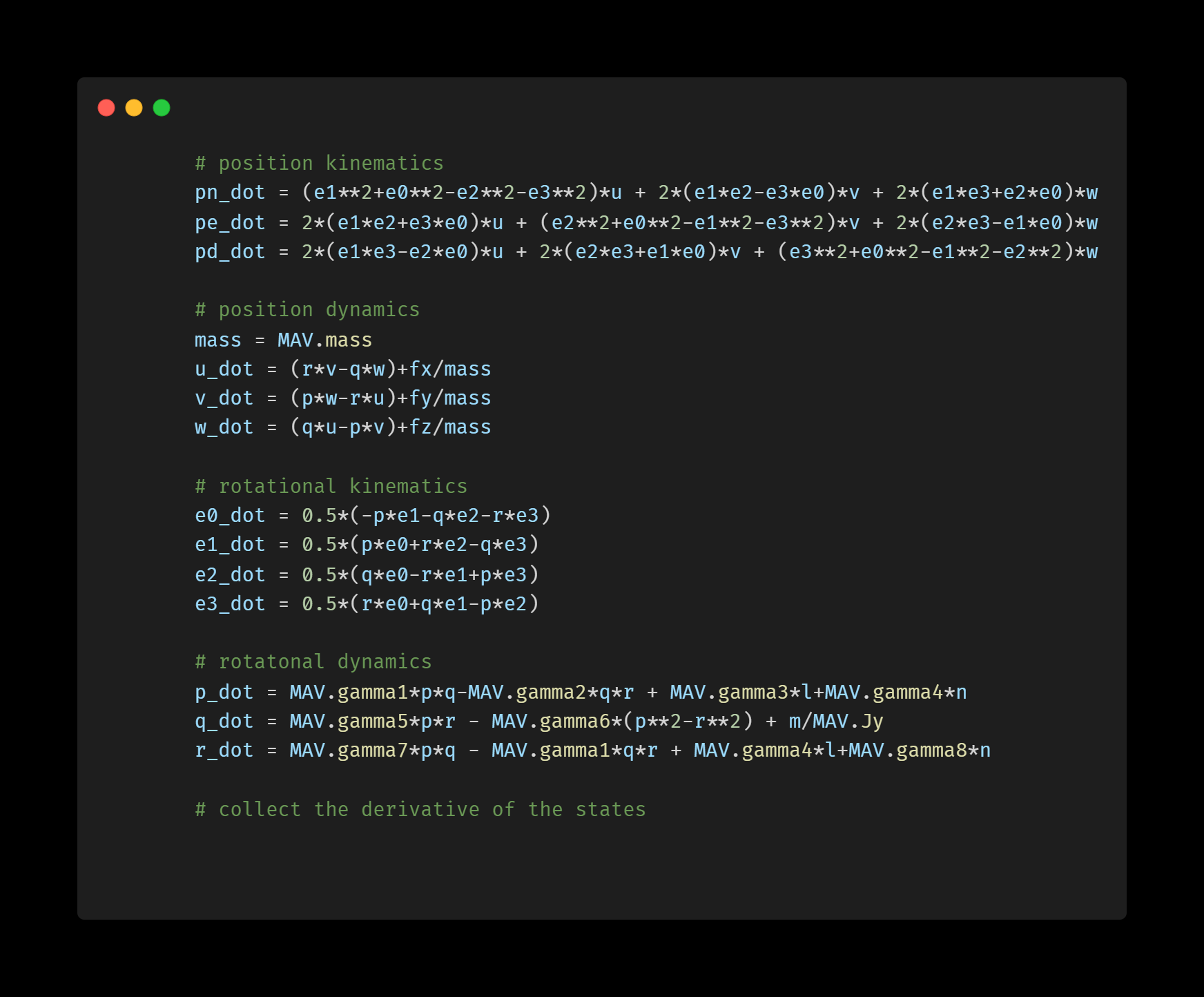
**(1)**

**(2)**

**(3)**

**(4)**

Python kodunu verilen eşitliklere göre düzenlediğimizde kod şekil 3.2’de gibi olmaktadır.

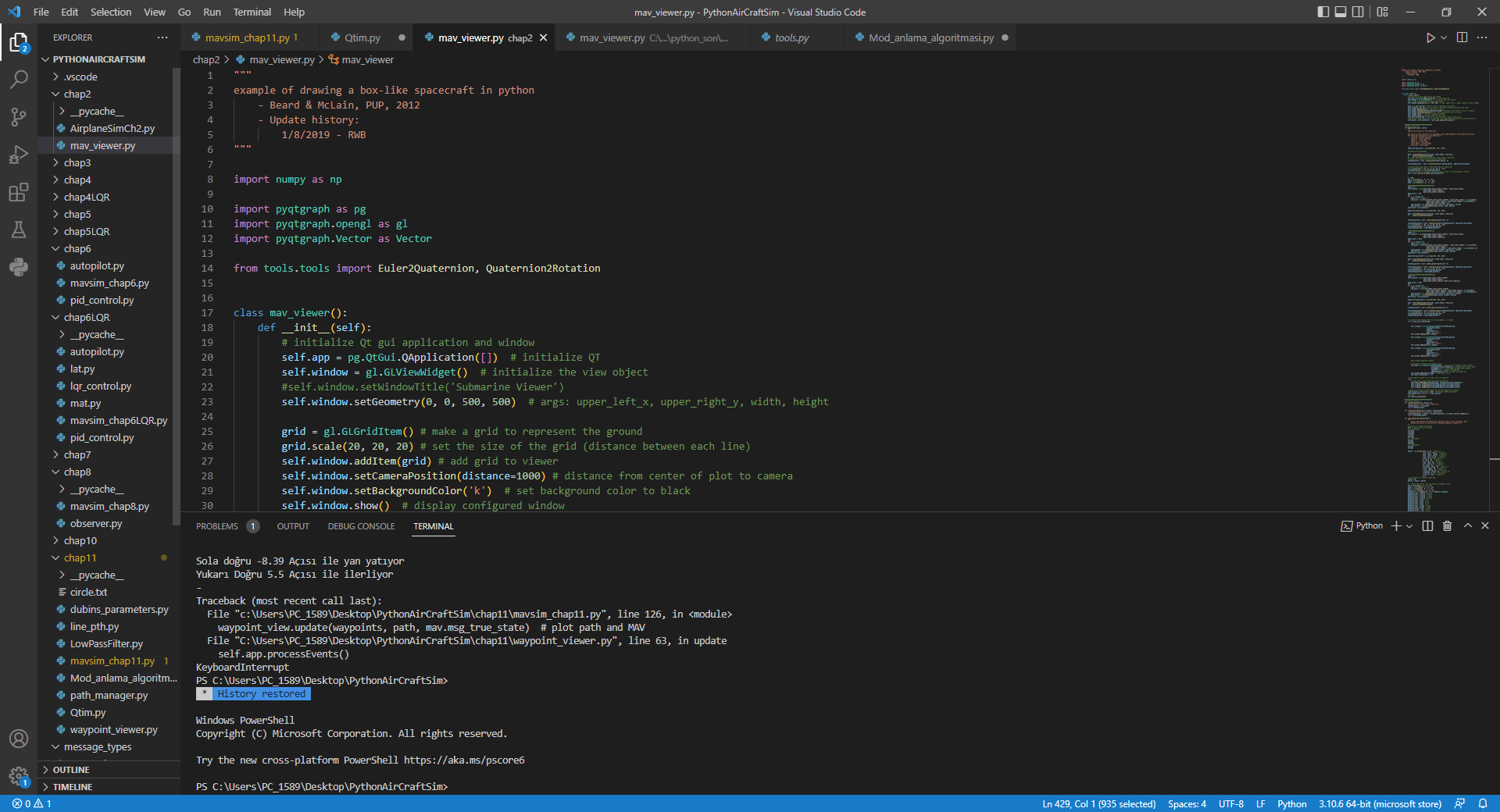


Şekil 3.2: Eşitlikler Yerleştirildiğindeki Kod Parçası

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

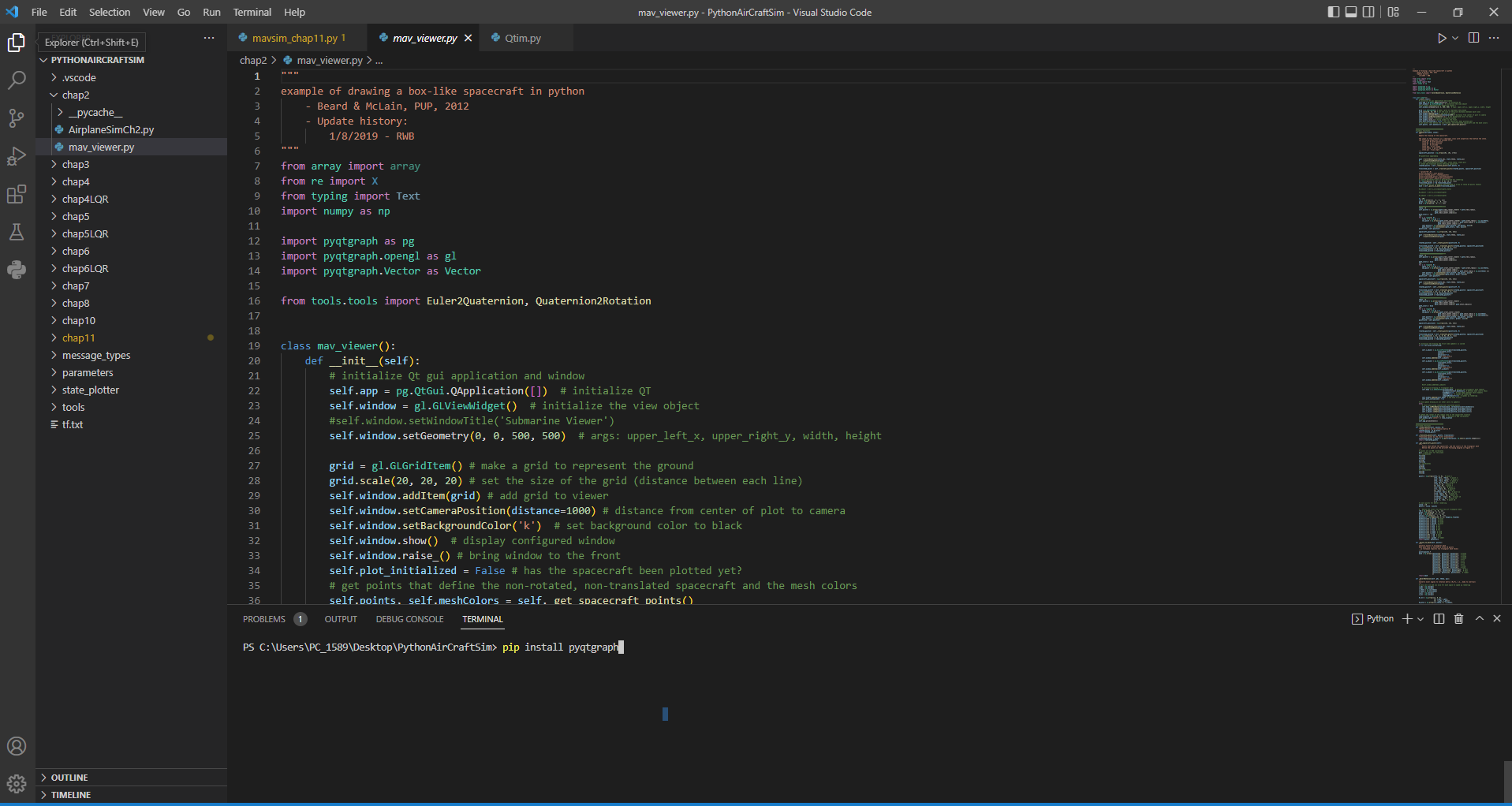
1. **Python Ortamında Gerekli Kütüphanelerin Kurulması**

Bu işlemler yapıldıktan sonra kaynak dosya çalıştırırken bazı kütüphaneler kurulmalıdır bu kütüphaneler kaynak dosyalarda şekil 4.1’de olduğu gibi en üste gösterilmektedir. Kodu derlemeye çalıştığınızda size eksik kütüphaneleri göstermektedir. Buradaki eksik kütüphaneleri bu şekilde belirleyebilir ya da dosyaların üst taraflarına bakabilirsiniz.



Şekil 4.1: Üstte Gösterilen Kütüphanelerden Bir Tanesi

Kütüphanelerin kurulma işlemi terminale “pip install pyqtgraph” yazarak yapılmaktadır “pyqtgraph” yerine hangi kütüphaneyi kurmak istediğinizi belirtmelisiniz şekil 4.2’de gösterilmektedir.



Şekil 4.2: Kütüphane Kurulumu

Kütüphanelerin kurulumu tamamlandıktan dosyaların birbirlerini görebilmesi için sys.path.append komutunu kullanmalısınız. Bu komut ile farklı konumlardaki dosyaların birlikte çalışmasını sağlayabilirsiniz şu anda oluşturulan sistem masaüstünde PythonAirCraftSim dosyası için bulunmaktadır. Kullanılan bilgisayarda bütün dosyalar Desktop\PythonAirCraftSim içinde olduğu için bu şekilde yazılmıştır kendi dosyanızın bağlantı yoluna göre ayarlanmalıdır şekil 4.3’de gösterilmektedir. Bazı bilgisayarlarda append parantezi içindeki “r” komutu kullanılmadığı görülmüştür. Eğer sistem çalışmamış ise r parametresini silebilirsiniz yine aynı şekilde sonuçlanırsa append komutu kullanımı araştırabilirsiniz.



Şekil 4.3: “append” Komutu

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.

1. **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 Eşitlik (5)’de gösterilmektedir.

**(5)**

Heading offsett değeri headingin önceki değerinden offsett kadar büyük ise düz rotadan cıktığı kabul edilmektedir. Buradaki offsett değeri araç rüzgar vb dış etkilerden ufak sapmalar yaşayabilmektedir bu sapmaları gözardı edebilmek için eklenmiştir.

* 1. **Daire çizme**

Bir hava aracının çizdiği yarıçapı belirlemek için aracın hızını ve pitch açısını bilmek gereklidir bu değerleri bildiğimizde eşitlik (6)’dan yararlanarak ne kadarlık bir yarıçap’da daire çizdiğini hesaplayabiliriz. Burada aracın yere göre hızı g kaldırma kuvveti ve yaptığı açıdır.

**(6)**

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. Offsett sayesinde aracın yaptığı açı değeri offsett değerinden büyük olduğunda bir dönüş yapmış kabul edilir.

* 1. **İrtifa kontrol**

Bunun öncelikle verideki gürültüyü sönümlemek amacıyla alçak geçiren filtre kullanılmıştır detayları 5.3.1’de anlatılmaktadır. Alçak geçiren filtrenin uygulandığı altitude değeri eşitlik (7)’deki koşula göre irtifa kazandığı ya da kaybettiği anlaşılmaktadır. Değerinin üstünde bir irtifa değişimi görüldüğünde bir çıkarım yapmamız gerektiğini göstermektedir aşağıdaki eşitlikte gösterilmektedir.

**(7)**

Bu koşul sağlandığında

Eğer ise irtifa kazanıyor diyebiliriz

Eğer ise irtifa kaybediyor diyebiliriz

h değeri yükseklik(irtifa) 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 8’de birinci derece bir alçak geçiren filtre denklemi verilmektedir.

**(8)**

Burada a değeri Alçak geçiren fitrenin frekansına bağlı olarak alınmalıdır. Temel mantık yeni hesaplanan değeri anında kabul etmeyip bunu zamana yaymaktır diyebiliriz. a değeri 0’a ne kadar yaklaşırsa alçak frekans sınırını o kadar aşağıya çekebiliriz. Örneğin a değeri 0.1 iken 100 hz filtreleniyorsa a değeri 0.05 iken 50 hz filtrelenir diyebiliriz. Özetlemek gerekirse gürültülü bir veri var frekans arttıkça değeri daha hızlı bir şekilde istenen değeri yakalayacak ama frekans azaldıkça a değerine bağlı olarak hesaplanan değere ulaşmakta bir o kadar zorlaşacak ve verinin istenen değere yaklaşması zorlaştığından aslında veriyi geçirmemiş olarak kabul edebiliriz. Aralarında eşitlik (9)’daki gibi bir oran bulunmaktadır.

**(9)**

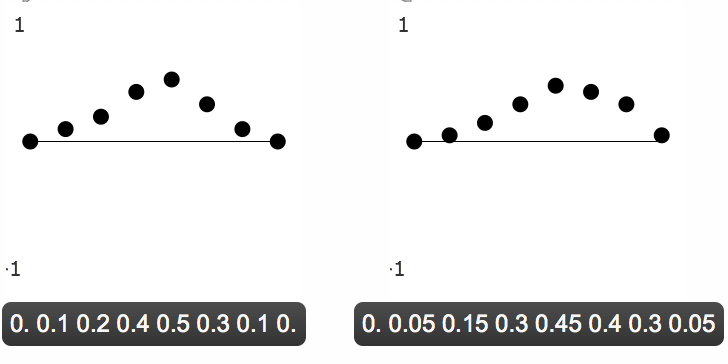
Grafik olarak göstermek gerekirse Şekil 5.3.1.1’deki grafikte görüldüğü üzere a değerini bağlı olarak istenilen frekans filtrelenebilir.

**Not:** (verilen örnekler direk olarak gerçeği yansıtmamaktadır sadece konunun anlaşılmasına yönelik verilmiştir. Burada örnekleme frekansı vb faktörlerde önemlidir.)

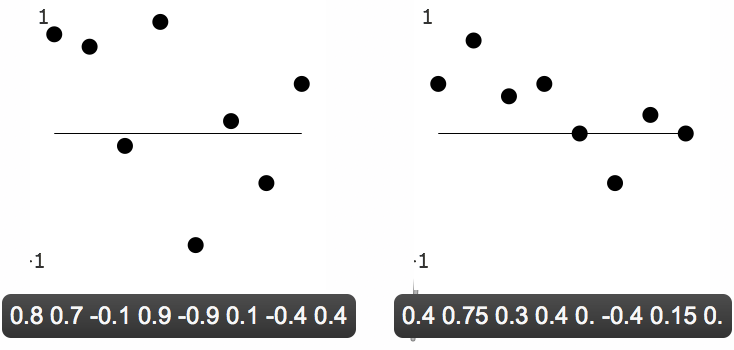
Şekil 5.3.1.1’deki grafik 0.01 aralıklarla ifade edildiği için sınırlı bir frekansa kadar filtreliyor gibi gözükmektedir. Grafik semboliktir ve gerçeği yansıtmamaktadır. Sadece a değeri ve frekans ilişkisini göstermek amacıyla yapılmıştır.

Şekil 5.3.1.1: Frekans ve a Değeri İlişkisi

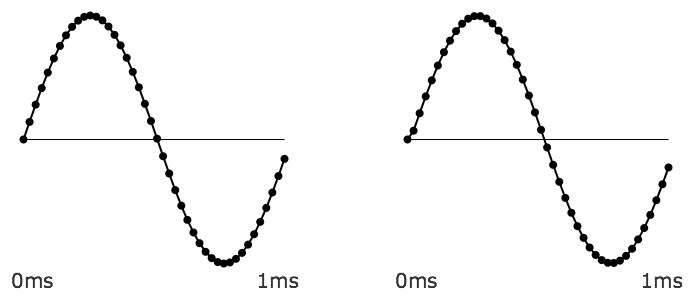
Şekil 5.3.1.2, Şekil 5.3.1.3, Şekil 5.3.1.4, Şekil 5.3.1.5’de a değeri 0.5 olarak ayarlandığında farklı girişlere verilen tepkiler ve bunlardan elde edilen çıkarımlar gösterilmektedir.



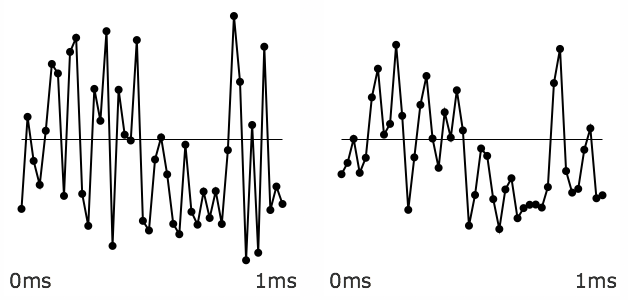
Şekil 5.3.1.2: Pürüzsüz bir giriş, alçak geçiren filtre tarafından yalnızca hafifçe düzleştirilir



Şekil 5.3.1.3: Pürüzlü bir giriş, alçak geçiren filtre tarafından daha belirgin şekilde yumuşatılır.



Şekil 5.3.1.4: Alçak geçiren filtreden neredeyse tamamen etkilenmeyen 1KHz sinüs tonu

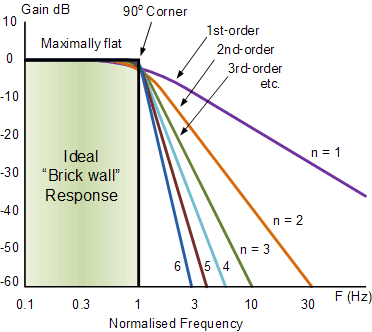


Şekil 5.3.1.5: Alçak geçiren filtre tarafından gürültülü bir giriş daha fazla değiştirilir*.*

Yukarıda anlatılan filtre birinci derece alçak geçiren bir filtredir. Eşitlik(9)’daki ifadede (1-a)’nın önündeki + ifadesini – yapıldığında yüksek geçiren bir filtre elde etmiş olursunuz.

**5.3.2 Alçak Geçiren Filtre dereceleri**

Filtrelerin dereceleri vardır ve bu dereceler arttıkça filtrenin frekans tepkisi artmaktadır. Şekil 5.3.2.1 filtrenin derecesine bağlı frekans tepkisini göstermektedir. Görüldüğü üzere filtrenin derecesi arttıkça kesim frekansında daha doğru sonuçlar vermektedir. Fakat Filtrelerin derecesinin artması bazı kötü özellikleri vardır.



Şekil 5.3.2.1: Filtre Derecesi ve Frekans Tepkisi

Bu kötü özelliklere Şekil 5.3.2.2’den bir çıkarım yapabiliriz. Görüldüğü üzere filtrelenen bir verinin 1. Ve 5. Derecede filtreler uygulandığında oluşan değişiklikleri göstermektedir.



Şekil 5.3.2.2: 2 Farklı Dereceden Filtrenin Karşılaştırılması

Değişikleri analiz edecek olursak filtrenin derecesi arttığında verinin Büyüklüğü (amplitude) düşmektedir ve zamanda geriye kaymaya başlamaktadır bunlarda istenmeyen durumlar ortaya çıkarmaktadır. Sinyalin genliğinin azalması sinyalin bozulmasına sebep olurken zamanda kayması da veriden oluşturacak çıktının gecikmesine sebep olmaktadır. Bu yüzden filtrelenecek veriye göre bu dereceler ayarlanmalıdır. Yapılan araştırmalarda hava araçlarında da genelde 2. Derece filtre kullanıldığı gözlemlenmiştir.

**Not:** İkinci derece low pass filtre yapmak isterseniz [2,3,4] linklerden yararlanabilirsiniz

**5.4 Yön Algılama:**

Yön algılama 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 sınırlarda yapıldığında rahatlıkla çıkmaktadır.

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

**5.5 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 göz ardı etmektir. Eşitlik (10)’da gösterilmektedir.

Roll merkezi 0 kabul edilmiştir.

**(10)**

Eşitlik (10)’daki denklem doğrulandığında

Eğer ise sağa doğru yatıyor değilse sola doğru yatıyor diyebiliriz. Pitch içinde aynı işlemleri yapabiliriz.

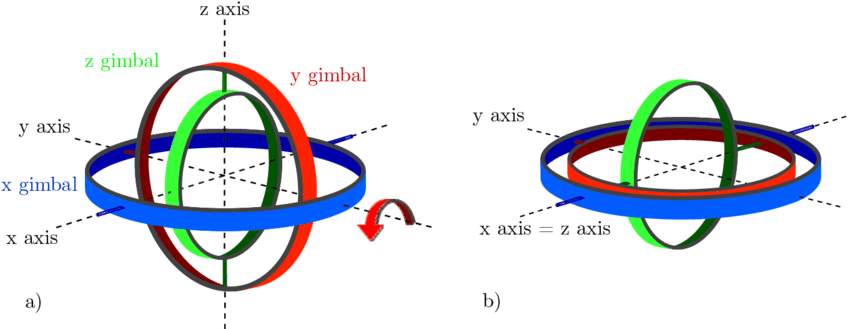
1. **Simülasyon eklentileri ve kullanımı:**

**6.1 Rotasyon Nedir Euler ve Quaternion Kullanımı**

Rotasyon kısa bir tabirle döndürme eylemi olarak tanımlanabilir. Cisimlerin merkezi etrafında yaptığı hareketlerin hepsine kapsamaktadır. Özellikle oyunlarda ve havacılıkta 3 boyutlu ortamda bir cismin dönüşünü ifade etmek için kullanılır. 2 farklı şekilde rotasyon yapılabilmektedir. Bunlar Euler ve Quanternion yöntemleridir temelde ikiside bir referans açı oluşturmak için kullanılabilir. Bu methodlar ve uygulanma şekilleri verilen dosyanın chap2 mav\_viewer.py dosyasında bulunmaktadır. Bu iki methodun bazı avantajları ve dezavantajlarından şu şekilde ifade edilebilir. Örneğin Euler insan beyninin daha rahat anlayıp yorumlayabilmesi için daha kolay bir metod olarak düşünülebilir. Quaternion ise insan beyninin algılaması ve yorumlaması oldukça zordur. Bu durumlar göz önüne alındığında Euler insanların anlaması bu kadar kolay iken neden bilgisayar ortamında Quanternion tercih ederiz ya da kullanırız.[9] Asıl sorun bilgisayara göre Quaternio’nun işlem maliyeti Euler göre daha azdır bunun sebeplerine gelicek olursak eğer Euler içindeki matris ve cosinüs işlemleri bir bilgisayar ve makine için hesaplama maliyeti daha zordur. Aynı zamanda Euler açısının bir sıkıntısıda Gimbal lock etkisidir.

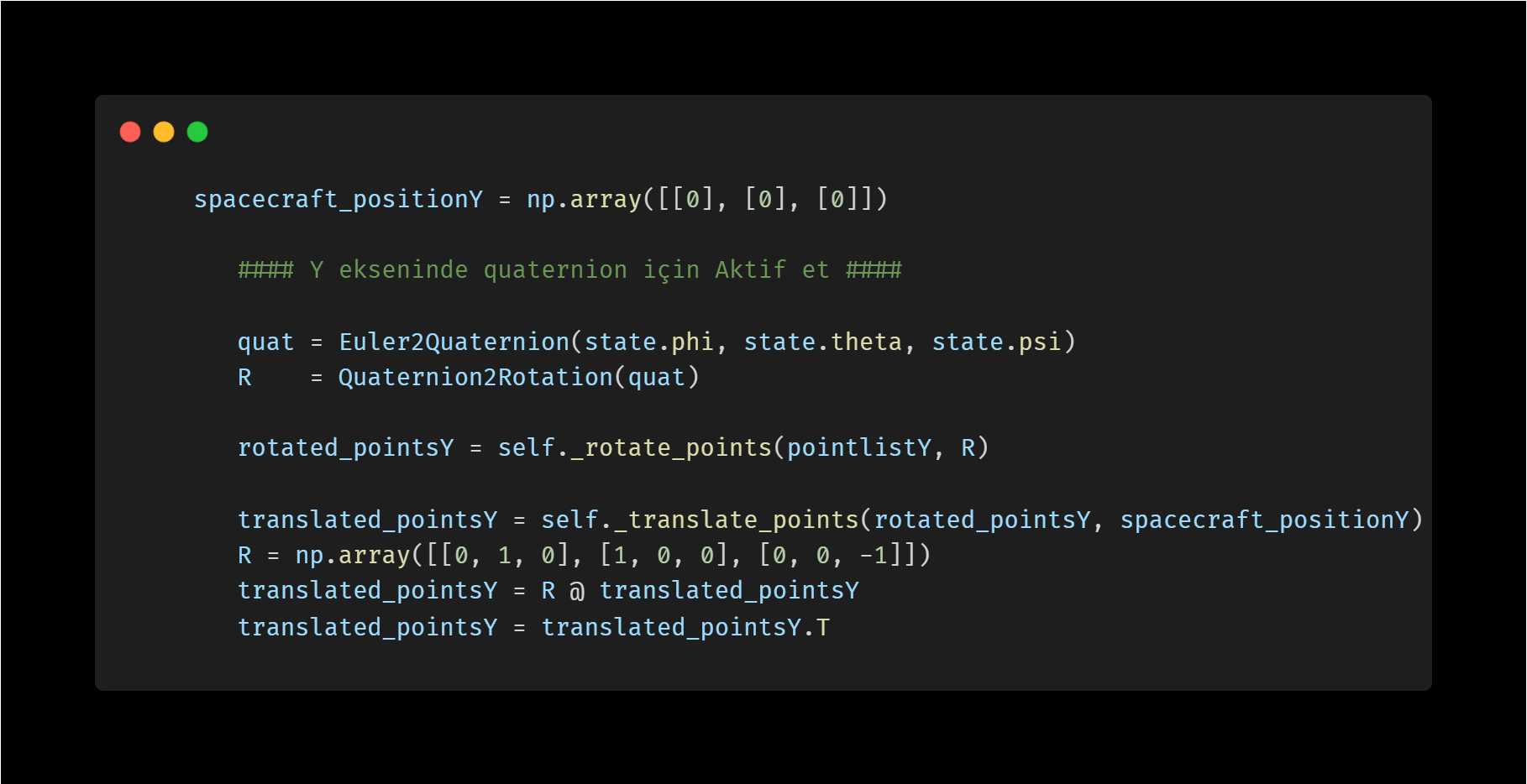
**6.1.1 Gimbal-Lock**

Üç gimbalin ikisinin eksenleri paralel bir konfigürasyona sürüldüğünde meydana gelen üç boyutlu bir uzayda bir derecelik açı kaybı , sistemi dejenere iki boyutlu bir uzayda dönmeye "kilitliyor". Şekil 6.1.1’de kırmızı ve mavi eksenlerin üst üste durduğu ve x eksenindeki hareketin ikisini de doğrudan etkileyeceği gözükmektedir. Apollo 11-13 görevinde bu sorun yaşanmıştır.[10].



Şekil 6.1.1: Gimbal-Lock Etkisi

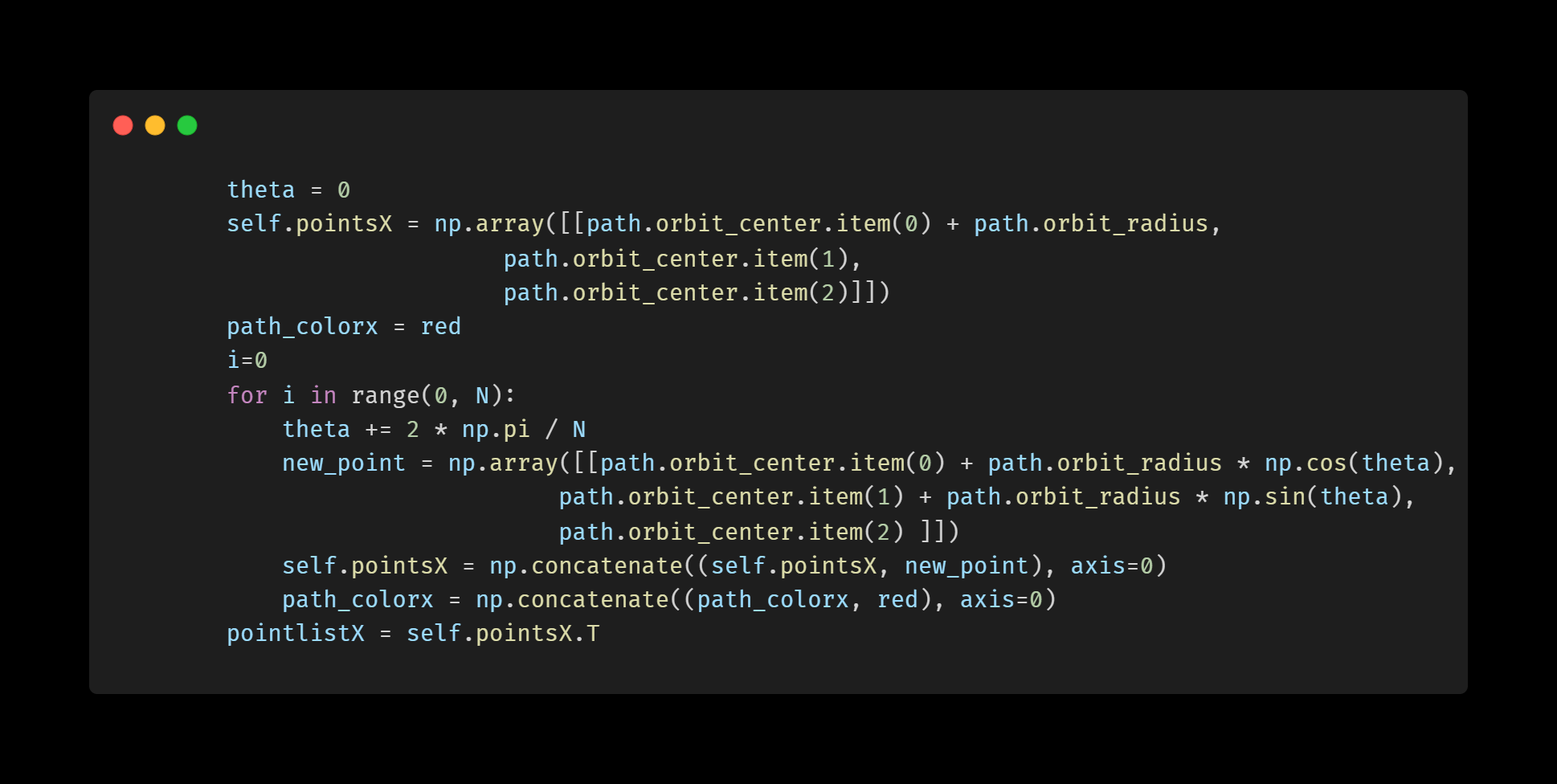
Yapılan çalışmanın Son halinde Quaternion kullanılmıştır. Euler açısı ile test yapılmak istenirse Euler açıları yorum olarak bulunmaktadır. Quanternion kapatılarak Euler açıları aktif hale getirilerek yapılabilir. Simülasyon ortamında bir cismi bir nokta üzerinden hareket ettirebilmek için 3 boyutlu ortamda cismin bütün noktalarını içeren bir matrisin olmalı ve bu matrisi hangi açıyla nereye hareket ettireceğini söyleyen Euler açılarının bulunması gerekmektedir. Şekil 6.1.2 simülasyonda kullanılan kod parçası gösterilmektedir. İlk olarak Euler Açılarını Quatenion’a çevirmek için Euler2Quaternion fonksiyonu kullanılmıştır. Daha sonra bu açılara bir R matrisine atılmıştır. R matrisi ile cismin koordinatlarını istenen açıda döndürmek için kullanmaktayız. “self.\_rotate\_points” fonksiyonundaki pointlistY oluşturulan cismin tanımlanan koordinatlarını temsil etmektedir. Konu 6.3’deki konuda tanımlanan koordinatların nasıl oluşturulduğu anlatılmaktadır. “self.\_translate\_points” fonksiyonundaki rotated\_pointsZ cismin açılara göre yaptığı koordinat değişimlerini içermektedir. spacecraft\_positionZ ise cismin dönüş yapacağı alanın yani döndürmek istediğin referans noktasının koordinatlarını içermektedir. Daha sonra Ned referans sistemine dönüştürmek için yeni bir R matrisi tanımlanıyor ve dönüştürme işlemi bittikten sonra şekli 3 boyutlu ortama aktarmak için matrisin transpozunu vermemiz gerekiyor bunun için en sonda transpozunu alıyoruz.



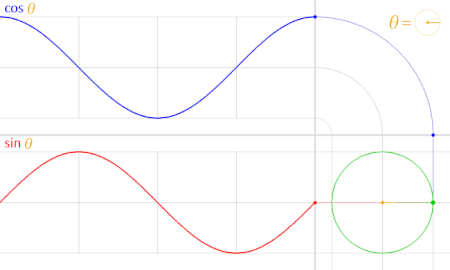
Şekil 6.1.2: Bir Cismi 3 Boyutlu Ortamda Döndürmek

**6.3 Üç Boyutlu Ortamda Daire Oluşturma**

Şekil 6.3.1’de bir dairenin matrisini oluşturmayı göstermektedir. Kod parçasını incelediğimizde ilk olarak tarama yapmak için başlangıç açısı 0 olan bir sabit theta belirledik. Daha sonra dairenin başlangıç noktasının belirlemek için self.pointsX adında y ve z de 0 olan x eksenine Radius kadar uzak olan bir nokta seçtik bu nokta bizim çizdirmek istediğimiz dairenin ilk noktasını göstermektedir. Bütün daireyi oluşturmak için bu ilk noktadan cosinüs ve sinüs fonksiyonu ile diğer noktalar oluşturulacak. “path\_colorx” ise matristeki noktaların hangi renkte çizdirileceğini belirtiyor “red” kod parçasında bulunan renk pigmentini tanımlayan bir 4’lü matris daha sonra for döngüsünde theta’yı 2 arttırarak daire çizdirebilirsiniz daire çizimini anlamak için kaynaklara bakabilirsiniz yada şekil 6.3.2 ye bakabilirsiniz [5]

****

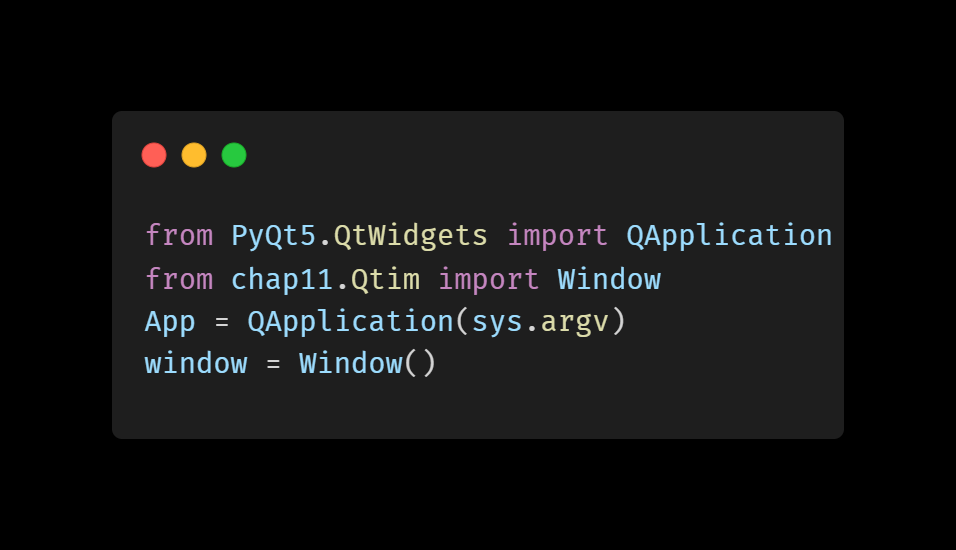
Şekil 6.3.1: İki Boyutlu Daire Çizdirme fonksiyonu



Şekil 6.3.2: İki Boyutlu Daire Cosinüs ve Sinüs Birlikte Çalışması

**6.2 Qtim Kütüphanesi fonksiyonları ve kullanımı:**

Qtim kütüphanesi yapılan işlerin hepsinin birleştirildiği ve birlikte çalıştırıldığı kütüphanedir. Qtim kütüphanesini diğer chapterlar’da kullanmak için aşağıdaki kod parçasını ilgili chapterın en üst tarafına eklemeniz ve şekil 6.2.1’deki window.update fonksiyonunu while döngüsüne eklemeniz yeterlidir.



Şekil 6.2.1 Qtim Kütüphanesi Kullanımı İçin Gerekli Fonksiyonlar

Qtim Kütüphanesi algoritmaları içermekle beraber bütün arayüz işlemleri ve layout yapısı UiComponents fonksiyonu içinde kullanılmaktadır. Arayüze yazı yazdırma 2 temel bölümden oluşmaktadır bunlardan bir tanesi Layout oluşturma ve label işlemleri. Layout oluşturma mantığı ile arayüz öncelikle bölümlere ayrılmış daha sonra içlerine labellar eklenerek yazdırma işlemleri yapılmıştır.[6][7] Kod parçasında açıklamaları mevcuttur. Arayüz olmadan çalıştırmak için chapter 11’de window.update fonksiyonun üstünde Mod\_Anlama.Update\_Mod aktif hale getirerek yapabilirsiniz.

Sistemi geliştirmek için yapılabilecekler ve şuan ki sorunlar:

* Eksenleri göstermek için dairelerin içlerine koordinat düzlemleri eklenebilir.
* Sistem arayüz ile birlikte çalıştığında kasmalar oluyor.
* Arayüz ile çalışırken 3 boyutlu ortam büyüyüp küçülüyor.

Kaynakça:

[1]<https://tr.wikipedia.org/wiki/Al%C3%A7ak_ge%C3%A7iren_filtre>

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

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

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

[5]<https://codepen.io/eanbowman/pen/xxzqJZ>

[6] <https://www.geeksforgeeks.org/pyqtgraph-clearing-the-line-in-line-graph/>

[7] <https://www.pythonguis.com/tutorials/pyqt-layouts/>

[8] <https://www.gamedeveloper.com/programming/rotating-objects-using-quaternions>

[9] <https://www.youtube.com/watch?v=0VAc_G79POE>

[10] <https://en.wikipedia.org/wiki/Gimbal_lock#:~:text=a%20flotation%20chamber.-,On%20Apollo%2011,to%20use%20a%20fourth%20gimbal>.