



Düşük Maliyetli 6 Eksen Hareketli Uçuş Simülatorü Prototip Tasarımı

Burak Hamdi Tufan^{1*}, Satılmış Ürgün²

^{1*} Kocaeli Üniversitesi, Havacılık ve Uzay Bilimleri Fakültesi, Havacılık Bilimi ve Teknolojileri Bölümü, Kocaeli, Türkiye, (ORCID: 0000-0002-4804-2010), burakhamditufan@gmail.com

² Kocaeli Üniversitesi, Havacılık ve Uzay Bilimleri Fakültesi, Havacılık Bilimi ve Teknolojileri Bölümü, Kocaeli, Türkiye (ORCID: 0000-0003-3889-6909), urgun@kocaeli.edu.tr

(İlk Geliş Tarihi 21 Ağustos 2020 ve Kabul Tarihi 11 Ekim 2020)

(DOI: 10.31590/ejosat.783420)

ATIF/REFERENCE: Tufan, B. H. & Ürgün, S. (2020) Düşük Maliyetli 6 Eksen Hareketli Uçuş Simülatorü Prototip Tasarımı (20), 299-310.

Özet

Yüksek doğrulukta işlemleri yapması beklenen uçuş simülatorleri için düşük maliyetli hareket sistemleri kullanılmaktadır. Bu sistemler daha kısa manipülatör, küçük hareket kapasitesi ve yüksek çalışma gürültüsü barındırmaktadır. Bu tarz etkilerin uçuş eğitimi alan pilotun algıları üzerindeki etkileri bilinmemektedir ve araştırılmaları gerekmektedir. Yüksek kalitedeki bir hareket platformu yaklaşımı ile düşük kapasiteli hareketin limitleri kaldırılabilir ve kapasitesi artırılabilir.

Yüksek güvenli hava taşımacılığında bir pilotun, tehlikeli hava şartlarındaki davranışlarının güçlendirilmesi gereklidir. Pilotların eğitimlerinde kullanılan simülatorler, bu tarz durumlarda uçağın vereceği tepkileri oluşturabilmelidir. Bu sayede pilotlar bu tarz durumlara karşı hazırlıklı olabilmektedir. Örneğin rüzgar kesmesi veya türbülans gibi durumlara karşı pilotlar hazırlıklı olmalıdır. Uçağın hareketi oldukça kısa bir süre içerisinde dengesizleşecektir. Bütün bu hareketleri, tasarlanan simülatorün oluşturabilmesi için güçlü dinamik ve dayanıklı manipülatörlere ihtiyacı olmaktadır. Gelişmiş bir kontrol algoritması geliştirebilmek için fiziksel yapının da limitlerini göz önüne almak gerekmektedir. Kullanılacak sabit platformun dayanımı, motorların sayısı ve tipi önemlidir. Manipülatör olarak bir tarafı zemine sabitlenen diğer tarafı hareketli mekanizmaya sabitlenen motorların kullanımlarıyla beraber, bir tarafı hareketli platforma sabitlenerek diğer tarafı döner motorun hareketli kısmına sabitlenen tasarımlar da mevcuttur. Her iki sisteminde hareket algoritmaları ve hesaplamaları değişecektir. Bu tarz fiziksel özellikleri ve limitleri algoritmaya uyarlamak gerekmektedir. Dolayısıyla algoritmaya uygulamanın dinamik modelini doğru ve etkili bir şekilde yansıtmak gerekmektedir. Çalışmanın ileriki aşamalarında artırılan bu kapasitelerin pilot adaylarının duyarlılığı üzerindeki etkileri araştırılabilir.

Bu çalışma Stewart platformu olarak bilinen bir 6DOF hareket platformu tasarım örneğidir ve her iki tarafı da hareketli bir noktaya sabitlenmiş manipülatörler ile daha etkili bir simülator hareket sistemi geliştirmeye yönelik çalışmalar yapılacaktır. Amacı Stewart platform tasarımlarının temel prensiplerine kılavuz olmak ile birlikte bu tasarımların ucuz ve hızlı bir şekilde prototipini oluşturmaya yardımcı olmaktır. Kinematik analizler gerçekleştirilerek küçük bir prototip üretilmiştir. Üretilen bu prototip stewart platformunun ilk olarak matematiksel modellenmesi oluşturulmuştur. Daha sonra bu modele uygun malzemeler seçilmiştir ve model seçilen bu malzemelere uygun hale getirilmiştir. Bu çalışma tamamen düşük bütçeli bir çalışmaya bağlı kaldığı için piyasadaki en ucuz fiyat-performans malzemeleri seçilmiştir. Ayrıca üretilen bu prototip için bir ara yüz yazılımı geliştirilerek gerçek uçuş bilgileri sağlanarak hesaplamaların ve analizlerin doğruluğu kontrol edilmiştir.

Bu çalışma daha gelişmiş bir uçuş simülatorü üretebilmek için gerçekleştirilen bir ön çalışmadır ve Kocaeli Üniversitesi | BAP Koordinasyon Birimi tarafından FYL-2020-2054 proje numarası ile desteklenmektedir.

Anahtar Kelimeler: Stewart platformu, uçuş simülatorü, 6DOF hareket, ters kinematik hesaplamaları, asenkron sistem ara yüzü

Low Cost 6 Axis Motion Flight Simulator Prototype Design

Abstract

Low cost motion systems are used for flight simulators that are expected to perform operations with high accuracy. These systems have shorter manipulators, small movement capacity and high operating noise. Such effects are unknown to the pilot's perceptions in

* Sorumlu Yazar: burakhamditufan@gmail.com

flight and need to be investigated. With a high quality motion platform approach, the limits of low capacity movement can be removed and its capacity can be increased.

In high security air transport, a pilot's behavior in hazardous weather conditions must be reinforced. The simulators used in the training of pilots should be able to create the reactions of the aircraft in such situations. In this way, pilots can be prepared for such situations. Pilots should be prepared for situations such as wind shear or turbulence, for example. The motion of the plane will become unstable in a very short time. In order to create all these movements, the designed simulator will need powerful dynamic and durable manipulators. In order to develop an advanced control algorithm, it is necessary to consider the limits of the physical structure. The strength of the fixed platform to be used, the number and type of motors to be used are important. Along with the use of motors as manipulator, one side fixed to the floor and the other fixed to the moving mechanism, there are designs with one side fixed to the moving platform and the other side fixed to the moving part of the rotary engine. Motion algorithms and calculations in both systems will change. It is necessary to adapt such physical properties and limits to the algorithm. Therefore, it is necessary to accurately and effectively reflect the dynamic model of the application to the algorithm. The effects of these increased capacities on the perceptions of pilot candidates can be investigated in the later stages of the study.

This study is an example of a 6DOF motion platform design known as the Stewart platform, and studies will be carried out to develop a more effective simulator motion system with manipulators fixed on both sides of a moving point. Its aim is to guide the basic principles of Stewart platform designs, as well as to help build prototypes of these designs cheaply and quickly. A small prototype was produced by performing kinematic analysis. The first mathematical modeling of this prototype Stewart platform was created. Later, suitable materials for this model were selected and the model was adapted to these selected materials. The cheapest price-performance materials in the market were chosen as this study was entirely dependent on a low-budget study. In addition, an interface software was developed for this prototype and the accuracy of the calculations and analysis was checked by providing real flight information.

This study is a preliminary study carried out to produce a more advanced flight simulator. It is supported by the Kocaeli Üniversitesi | BAP Coordination Unit with the project number FYL-2020-2054.

Keywords: Stewart platform, flight simulator, 6DOF motion, inverse kinematic calculations, asynchronous system interface

1. Giriş

Stewart platformu tasarımları Stewart'ın çalışmasından sonra hareketli tasarımlar için çok popüler bir konu haline gelmiştir[1]. Stewart platformlarının alt sistemleri ile ilgili birçok araştırma yapılmıştır. Bu alt sistemlerden bazıları hareket sistemi, konum algılama, kinematik, mekanik ve dinamik gibi pek çok analiz çalışmaları yapılmıştır. Stewart platformlarının uygulama alanları yüksek doğruluk ve seri manipülatörlerinden gelen yüksek kuvvetinden dolayı geniştir. Özellikle Havacılık-Uzay çalışmaları için gerekli pilotaj eğitimlerinde[2], engebeli araziler için tasarlanan askeri mobil gözetim robotlarında[3] sıkça karşımıza çıkmaktadır. Ayrıca sağlık alanında da çalışmalar yapılmaya başlanmıştır[4].

Uçuş simülatörleri bütün dünyada pilot eğitimleri için kullanılan etkili ve güvenli eğitim araçlarıdır. Bu sayede pilot adayları uçuş esnasında meydana gelebilecek olan durumlara karşı manevra ve prosedür eğitimi almaktadır. Bununla beraber pilot eğitimleri için hareket simülatörleri kullanımının verimliliği hakkında bir çalışma mevcuttur [5]. Yakın zamanda yapılan bir analiz çalışmasında hareketli simülatörlerin manevra pratiği konusunda yeni başlayan pilotlar için önemli olduğunu fakat bu alanda deneyimli pilotlar için çok önemli olmadığını ortaya koymuştur [6]. Bu şekildeki karakteristikler üzerinde yapılan çalışmalardan dolayı hareketli simülatörlerde genel bir sonuca varmak pek mümkün değildir.

Herhangi bir simülatör sadece en yüksek doğrulukta bir eğitim aracı olarak kabul edilebilir. Bu durum simülatörün, simüle ettiği aracın gerçek dünyadaki davranışlarını yüksek doğrulukta gerçekleştirmesi gerekmektedir. Bu, beceriye dayalı davranışın tanımlanmasıyla [7] ve bir pilot modelin tanımlanmış parametrelerindeki değişikliklerin değerlendirilmesiyle [8] objektif olarak ölçülebilir. Simülatör hareketinin etkisi üzerine yapılan çalışmalar, gerçekleştirilen kapalı döngü kontrol görevlerinde pilot davranışında önemli değişiklikler olduğunu göstermiştir [9]. Benzer şekilde, simülatör hareketi

kullanıldığında, simülatör hareketinin kapatıldığı koşullara kıyasla pilot performansında bir artış bulunmuştur [10].

Birçok uçuş simülatörü çalışmasında hareket mekanizması olarak sabit bir zemine altı adet doğrusal olarak uzayabilir manipülatör ile bağlanan hareketli bir zeminden oluşan Stewart platformu kullanılmaktadır [11]. Bu çalışmalarda genellikle hareket platformlarının kinematik hesaplamalarına [12] ve bu hesaplamalardan çıkan sonuçların her bir manipülatöre nasıl uygulanacağı [13] konuları üzerine yoğunlaşmıştır. Bu hareketler için gerekli olan ters kinematik hesaplamalar hareket parametreleri doğrusal olarak değişmeyen sistemlerin kapalı sistem kontrolü için gerekmektedir [14]. Bu tasarımda bir uçuş simülatörü prototipi olduğu için dışarıdan alınan anlık uçuş verilerine göre hareket parametreleri değişecektir. Dolayısıyla hareket değerleri doğrusal olmayacaktır. Bu yüzden ters kinematik hesaplamaları ile bir kapalı sistem kontrolüne ihtiyaç duyulmaktadır.

Stewart platformunun hareketlerinin doğruluğu için gereken anlık hesaplama gücünün yüksek olmasından dolayı son 20 yılda işlemci teknolojisinin büyük yol almasına bağlı olarak gereken hesaplama gücü karşılanabilmiş ve Stewart platform çalışmaları hızlanmıştır. Hazırlanan Stewart platformlarının yüksek doğruluk ile hareket edebilmesi için hazırlanan matematiksel modele ve bu model için oluşturulan matematiksel hesaplamalara tam olarak uygun olması gerekmektedir. Aksi takdirde platforma bağlı olan manipülatörler, rotlar, hareketli platform sistemleri biri veya birkaçı birden zarar görebilir. Ayrıca bu duruma bağlı olarak kullanıcıya da zarar verebilir. Ayrıca üretilen tasarımın, matematiksel modele daha uygun olabilmesi için CAD dizayn araçlarından yararlanılması doğruluğu ve güvenliği arttırmaktadır.

Bu çalışmada çok düşük bir maliyet ile üretilecek olan bir Stewart platformu tasarlanmıştır. Tasarlanan bu Stewart platformunda düşük maliyetli işlemci kullanılmıştır. Bu işlemci ARM Cortex M0 mimarisi üzerine kurulu STM32F051 olarak seçilmiştir. Sonraki bölümde temel olarak küçük boyutlu

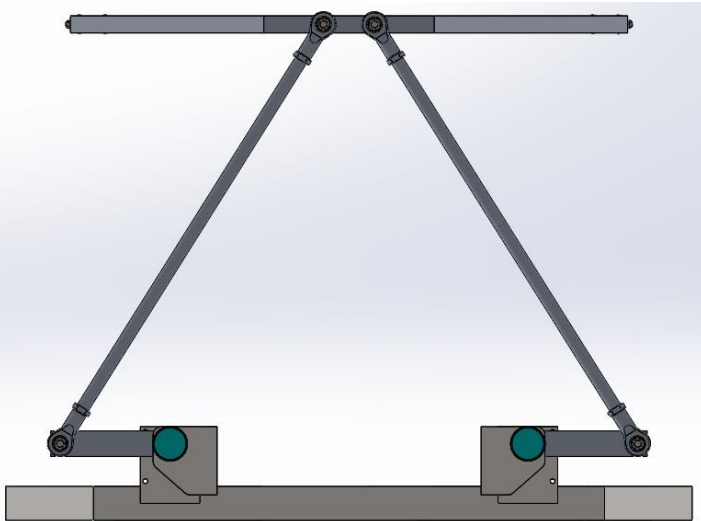
prototip bir uçuş simülâtörü ve bu çalışmanın gerekliliklerini hakkında bilgi verilecektir. 3. bölümde bu tasarımın matematiksel özellikleri ve modellemesi ile ilgili bilgiler verilecektir. 4. bölümde ise oluşturulan matematiksel modelin hesaplama yapabilmek için gereken kodlamalar yapılmıştır. 5. bölümde çalışmanın sonuçları hakkında bilgiler verilecek ve çalışma ile ilgili önerilerde bulunulacaktır.

2. Prototip 6 Eksenli Stewart Platformunun Tasarım Kriterleri

Bu bölümde üretilecek olan düşük maliyetli 6 eksen hareketli uçuş simülâtör analizi yapılacaktır. Bu prototipin üretimi, kullanılan malzemelerin matematiksel özellikleri ve bu malzemelerin CAD uygulamalarında kullanımları gösterilecektir. Aynı zamanda örnek uçuş verileri için Microsoft tarafından geliştirilen Flight Simülâtör uygulamasından faydalanılacak ve üretilecek olan simülâtöre çalışma kapsamında geliştirilen bir arayüz aracılığıyla seri port haberleşme protokolü kullanılarak örnek uçuş verisi göndererek hareket etmesi sağlanacaktır. Tasarım tamamen düşük maliyetli olduğu için piyasadaki en ucuz malzemeler kullanılacaktır. Toplam fiyatı 50 USD altında olan bu çalışmanın bazı parçaları CAD programında hazırlanan model ile boyut ve bağlantı noktaları aynı olacak şekilde üretilmiştir.

Tasarım 6 adet manipülâtörden oluştuğu için 6 tane mini servo motor kullanılmıştır. Servolar ile rotaları hareket ettirebilmek için tasarlanan ve 3 boyutlu yazıcı ile üretilen kollar kullanılmıştır ve rotaları kollara bağladıktan sonra kolların uygun yerlerini servoların hareketli uçları bağlanmıştır. Servolar platformun sabit kısmına; rotaların diğer uçlarını ise hareketli platform üzerinde belirlenen noktalara sabitlenmiştir.

Şekil-1'dek bir motor çiftinin iki boyutlu bağlantısı görülmektedir. Tasarımda bağlantısı yapılmış olan 3 adet motor çifti bulunmaktadır ve her biri aşağıdaki şekildeki gibi bağlanmıştır. Bağlantı şekillerinde konumların tam olarak belirlenmesi konum hesaplamaları için çok önemlidir. Bu yüzden eşit aralıklarla olmasına dikkat edilmiştir ve her biri merkeze göre 120° derecelik aralıklarla konumlandırılmıştır. Aynı şekilde hareketli platform bağlantı çiftleri de 120° derecelik aralıklarla konumlandırılmıştır. Platformun simetrik olmasına dikkat edilmiştir.



Şekil 1: Motor çifti bağlantısını gösteren teknik çizim

Tasarım özelliklerinin aktarılacağı işlemci seçilirken geliştirme ortamı ve programlama dili de seçilmelidir. Bu çalışmada gömülü sistem tarafında STM32F051 mikroişlemcisi için uVision Keil derleyicisinde C dili kullanılmıştır. Yine C++ dili ve QT Geliştirme ortamı kullanılarak geliştirilen arayüz uygulaması ile MS Flight Simülâtör programındaki anlık uçuş verilerini seri port üzerinden Stewart platform prototipine gönderen arayüz uygulaması geliştirilmiştir.

Bu çalışmada izlenebilecek iki yöntem bulunmaktadır. İlk yöntemde; hesaplamaları arayüz programında gerçekleştirerek gerekli motor açılarını gömülü sisteme göndermek ve gömülü sistemin sadece motor açılarından sorumlu olmasını sağlayabilmektir. Bu şekilde işlem hızı düşük olan işlemcilerdeki yükü daha yüksek işlem kapasiteli olan arayüz uygulamasına verebilmek mümkün olmaktadır. Diğer yöntemde ise uçuş programından alınan hedef roll, pitch ve yaw değerlerin gömülü sistem tarafına göndererek hesaplamaların gömülü tarafta gerçekleşip motorları sonuç açısına getirip o değerde tutmaktır. Bu yöntem düşük hızlı işlemcilerde tavsiye edilmemektedir. Çünkü kinematik hesaplamalar için hızları yetmeyecek ve harekette problemler ortaya çıkacaktır. Seçilen işlemcinin 48MHz frekansa sahip olması ile birlikte haberleşme ve PWM sinyalinin eş zamanlı olarak farklı donanımsal katmanlardan yapması tasarım için yeterli olmaktadır. Dolayısıyla bu çalışmada ikinci yöntem tercih edilecek ve gerekli işlemler gömülü tarafında yapılacak. Arayüz uygulaması sadece MS Flight Simulator 'den verileri okuyup seri port üzerinden gönderecektir. Bu sayede MS Flight simülâtör uygulamasındaki uçağın yatış değerleri anlık olarak değiştiğinde gömülü sistem çok hızlı bir şekilde haberleşip kinematik hesaplamaları yaptıktan sonra PWM sinyallerini motorlara çok hızlı bir şekilde aktaracaktır.

Bu bölümde matematiksel modellemenin bu çalışma ile ilgili öneminden bahsedilecek ve bu işlemlerin nasıl yapıldığı anlatılacak. Matematiksel modelleme bir çalışmanın temelidir[15]. İçerisinde çok fazla matematiksel hesaplama barındıran bir ürünün, doğru bir şekilde matematiksel modellemesi oluşturulmadan başlanırsa bir süre sonra tasarımsal olarak ciddi problemler meydana gelebilmektedir. Çünkü netlik gerektiren bir çalışma, belirsizlikler üzerinden devam ettirilmeye çalışılmış olacaktır.

Bu tasarımda da çok fazla matematiksel hesaplama bulunması nedeniyle ilk olarak tasarım kriterlerinin belirlenmesi gerekir. Sonrasında bu değerler bir CAD programı aracılığı ile üç boyutlu tasarıma aktarılmış ve sistemin üç boyutlu görüntüsü oluşturulmuştur. Oluşan bu üç boyutlu görüntü ile sistemde olağandışı ve olmaması gereken durumlar gözlemlenmiştir. Oluşmuş olan bu tarz durumları ortadan kaldırmak için yeniden gerekli değişiklikler yapılarak sistemin uygun duruma gelmesi sağlanmıştır. Bu tarz kontrollerin ardından sistemin hareketleri için gerekli matematiksel hesaplamalar oluşturulmuş ve bu hesaplamaları gerçekleştiren bilgisayar programı yazılmıştır. Son olarak bu hesaplama sonuçlarını işlemciye aktarma protokolü ve sistemi hazırlanmıştır.

3. Platformun Matematiksel Modellemesi

6DOF hareket sistemlerin tasarımında Tait-Bryan ve Euler matris dönüşüm teoremleri yardımcı olmaktadır[16]. Bu teoremler ile bir noktanın 3 boyutlu eksen üzerindeki hareketi hesaplanabilmektedir. Bu hesaplamalar bütün platform

bağlantıları için yapıldığı zaman bir sistemin 6 eksen üzerindeki hareketi merkez noktaya göre hesaplanmış olmaktadır.

Üç boyutlu ekseninde bir nokta ele alınsın ve bu nokta hareket ettirilsin. Hareketin sonunda gelineyen yeni noktanın merkeze olan açıları X,Y ve Z eksenlerine göre sırasıyla α , β ve ϕ açıları kadar döndürülmüş olsun. Sonuç olarak yeni noktalar eski durum göz önünde bulundurularak merkeze göre $X+\alpha$, $Y+\beta$ ve $Z+\phi$ kadar hareket etmiş olacaktır. Trigonometrik fonksiyonlar ile yapılan hesaplamalar aşağıdaki gibidir.

Eksenler etrafındaki dönüşleri hesaplayabilmek için ilk olarak aşağıdaki trigonometrik dönüşüm fonksiyonlarının uygulanması gerekmektedir.

$$\cos(a + b) = \cos a \cos b - \sin a \sin b \quad (1)$$

$$\sin(a + b) = \sin a \cos b + \cos a \sin b \quad (2)$$

Tait-Bryan teoremine göre dönüş hareketleri ilk olarak Z eksenini etrafında, daha sonra Y eksenini ve en son X eksenini etrafında yapılmaktadır. Her bir dönüş için eksenlerdeki açıya bağlı uzaklık değişimleri incelenecek ve ilgili eksenler üzerindeki hareket hesaplanacaktır.

Bir noktanın üç eksenindeki hareketi aşağıdaki hesaplamalarla yapılmaktadır.

$$R_z = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r * \cos(Z + \phi) \\ r * \sin(Z + \phi) \\ d_z \end{bmatrix} \quad (3)$$

$$= \begin{bmatrix} r * (\cos(Z) * \cos(\phi) - \sin(Z) * \sin(\phi)) \\ r * (\cos(Z) * \sin(\phi) + \sin(Z) * \cos(\phi)) \\ x * 0 + y * 0 + z * 1 \end{bmatrix}$$

Matris içerisindeki parantezler açıldığında aşağıdaki sonuçlar elde edilmektedir.

$$R_z = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r * \cos(Z + \phi) \\ r * \sin(Z + \phi) \\ d_z \end{bmatrix} \quad (4)$$

$$= \begin{bmatrix} r * \cos(Z) * \cos(\phi) - r * \sin(Z) * \sin(\phi) \\ r * \cos(Z) * \sin(\phi) + r * \sin(Z) * \cos(\phi) \\ x * 0 + y * 0 + z * 1 \end{bmatrix}$$

Bu eşitliklerde Z açısı z eksenini etrafındaki dönüş hareketi olmadan önceki açı değeridir. Bu değerler ile başlangıç durumlarında eksenlere olan uzaklık hesaplanabilir. Dolayısıyla bu dönüş için $x = r * \cos(Z)$ ve $y = r * \sin(Z)$ olacaktır. Bu bilgilerle eşitlik yeniden düzenlendiğinde aşağıdaki eşitlikler ile bu eksen etrafındaki dönüş sonrası konum elde edilecektir.

$$R_z = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r * \cos(Z + \phi) \\ r * \sin(Z + \phi) \\ d_z \end{bmatrix} \quad (5)$$

$$= \begin{bmatrix} x * \cos(\phi) - y * \sin(\phi) + z * 0 \\ x * \sin(\phi) + y * \cos(\phi) - z * 0 \\ x * 0 + y * 0 + z * 1 \end{bmatrix}$$

$$R_z = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (6)$$

Yukarıdaki işlemlerin aynısını Y eksenini etrafındaki dönüş için yapıldığında bu kez Y eksenine olan uzaklık değişmeyecektir. Dolayısıyla değişimler X ve Z eksenlerinde olacaktır.

$$R_y = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r * \cos(Y + \gamma) \\ d_y \\ r * \sin(Y + \gamma) \end{bmatrix} \quad (7)$$

$$= \begin{bmatrix} r * (\cos(Y) * \cos(\gamma) - \sin(Y) * \sin(\gamma)) \\ x * 0 + y * 1 + z * 0 \\ r * (\cos(Y) * \sin(\gamma) + \sin(Y) * \cos(\gamma)) \end{bmatrix}$$

Matris içerisindeki parantezleri açıldığında aşağıdaki sonuç elde edilecektir.

$$R_y = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r * \cos(Y + \gamma) \\ d_y \\ r * \sin(Y + \gamma) \end{bmatrix} \quad (8)$$

$$= \begin{bmatrix} r * \cos(Y) * \cos(\gamma) - r * \sin(Y) * \sin(\gamma) \\ x * 0 + y * 1 + z * 0 \\ r * \cos(Y) * \sin(\gamma) + r * \sin(Y) * \cos(\gamma) \end{bmatrix}$$

Bu eşitliklerde Y açısı y eksenini etrafındaki dönüş hareketi olmadan önceki açı değeridir. Bu değerler ile başlangıç durumlarında eksenlere olan uzaklık hesaplanabilir. Dolayısıyla bu dönüş için $x = r * \cos(Y)$ ve $z = r * \sin(Y)$ olacaktır. Bu bilgilerle eşitlik yeniden düzenlendiğinde aşağıdaki eşitlikler ile bu eksen etrafındaki dönüş sonrası konum elde edilecektir.

$$R_y = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r * \cos(Y + \gamma) \\ d_y \\ r * \sin(Y + \gamma) \end{bmatrix} \quad (9)$$

$$= \begin{bmatrix} x * \cos(\gamma) + y * 0 - z * \sin(\gamma) \\ x * 0 + y * 1 + z * 0 \\ x * \sin(\gamma) + y * 0 + z * \cos(\gamma) \end{bmatrix}$$

$$R_y = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(\gamma) & 0 & -\sin(\gamma) \\ 0 & 1 & 0 \\ \sin(\gamma) & 0 & \cos(\gamma) \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (10)$$

Son olarak x eksenini etrafındaki dönüş aşağıdaki gibi olacaktır.

$$R_x = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} d_x \\ r * \cos(X + \omega) \\ r * \sin(X + \omega) \end{bmatrix} \quad (11)$$

$$= \begin{bmatrix} x * 1 + y * 0 + z * 0 \\ r * (\cos(X) * \cos(\omega) - \sin(X) * \sin(\omega)) \\ r * (\cos(X) * \sin(\omega) + \sin(X) * \cos(\omega)) \end{bmatrix}$$

Matris içerisindeki parantezleri açıldığında aşağıdaki sonuç elde edilecektir.

$$R_x = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} d_x \\ r * \cos(X + \omega) \\ r * \sin(X + \omega) \end{bmatrix} \quad (12)$$

$$= \begin{bmatrix} x * 1 + y * 0 + z * 0 \\ r * \cos(X) * \cos(\omega) - r * \sin(X) * \sin(\omega) \\ r * \cos(X) * \sin(\omega) + r * \sin(X) * \cos(\omega) \end{bmatrix}$$

Bu eşitliklerde X açısı x eksenine etrafındaki dönüş hareketi olmadan önceki açı değeridir. Bu değerler ile başlangıç durumlarında eksnelere olan uzaklık hesaplanabilir. Dolayısıyla bu dönüş için $y = r * \cos(X)$ ve $z = r * \sin(X)$ olacaktır. Bu bilgilerle eşitlik yeniden düzenlendiğinde aşağıdaki eşitlikler ile bu eksen etrafındaki dönüş sonrası konum elde edilecektir.

$$R_x = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} d_x \\ r * \cos(Y + \omega) \\ r * \sin(Z + \omega) \end{bmatrix} \quad (13)$$

$$= \begin{bmatrix} x + y * 0 + z * 0 \\ x * 0 + y * \cos(\omega) - z * \sin(\omega) \\ x * 0 + y * \sin(\omega) + z * \cos(\omega) \end{bmatrix}$$

$$R_x = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega) & \sin(\omega) \\ 0 & \sin(\omega) & \cos(\omega) \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (14)$$

Noktanın üç eksen etrafındaki hareketinin hesaplanabilmesi için yukarıdaki 3 tane dönüş matrisi sırasıyla çarpılacak. Çıkan matris ile cismin 3DOF hareketini verecektir.

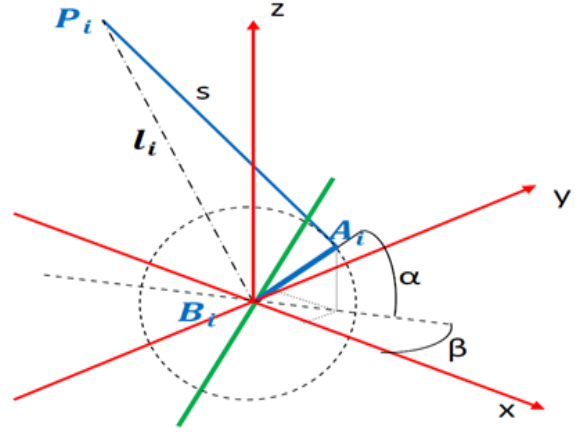
$$R = R_z(\phi) * R_y(\gamma) * R_x(\omega)$$

$$R = \begin{bmatrix} \cos(\gamma) \cos(\phi) & \sin(\omega) \sin(\gamma) \cos(\phi) - \cos(\omega) \sin(\phi) & \cos(\omega) \cos(\phi) \\ \cos(\gamma) \sin(\phi) & \sin(\omega) \sin(\gamma) \sin(\phi) + \cos(\omega) \cos(\phi) & \cos(\omega) \sin(\phi) \\ -\sin(\gamma) & \sin(\omega) \cos(\gamma) & 0 \end{bmatrix}$$

Şu ana kadarki hesaplamalarda platformun roll, pitch ve yaw tanımlanmıştır. Fakat bu tasarımdan beklenen hareket kabiliyeti 6DOF olarak belirlendiği için bu matrise eksenler üzerindeki kayma hareketleri eklenecektir. Bu sayede platform eksenler üzerinde kayma hareketlerini de gerçekleştirebilecektir. Bu hareketler sağ-sola, yukarı-aşağı ve ileri-geri hareketleridir. Bu

hareketler dönüş hareketinden bağımsız olduğu için yukarıdaki denkleme kayma değişkenleri ayrıca eklenmelidir.

Bu yüzden yukarıdaki matris yardımı ile her bir platform rot bağlantısının konumlarını hesaplanacaktır. Sonrasında kayma matrisini ekleyerek platformun yüksekliğini hesaplanacaktır. Dolayısıyla başlangıç durumdaki her bir manipulator için pozisyon bilgisi aşağıdaki gibi olacaktır[17].



Şekil 2: Servo motorun x-y eksenleri üzerindeki konumlandırılması

Şekil 2'ye göre:

P_i : Platformun Rod ile bağlandığı noktalar. Vektörü : p_i

B_i : Servoların Base bağlantı noktaları. Vektörü : b_i

A_i : Servo kolu ile rodun bağlantı noktaları. Vektörü : a_i

r : Rod uzunluğu

l_i : B_i ile P_i noktaları arasındaki vektör

Her bir manipülatör için yukarıda belirtilen konum vektörleri ise aşağıdaki gibi ifade edilir.

$$p_i = [x_{pi} \ y_{pi} \ z_{pi}]^T \quad (17)$$

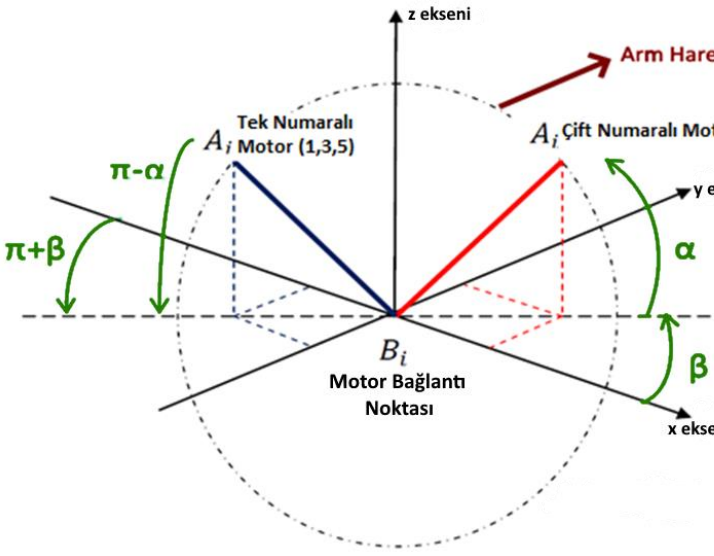
$$b_i = [x_{bi} \ y_{bi} \ z_{bi}]^T \quad (18)$$

$$a_i = [x_{ai} \ y_{ai} \ z_{ai}]^T \quad (15) \quad (19)$$

Bu tasarımda motorlar çift olarak ele alınmaktadır. Motor çiftlerinin dönüş yönleri birbirlerine ters olacak şekilde ayarlanmıştır. Dolayısıyla motorların dönüş yönlerini her iki yöne dönüşlerini belirtmek adına numaralandırırken tek ve çift numaralı motorlar olarak isimlendirilecektir. Ayrıca motorların dönüşleri kendi eksenlerine göre "y" eksenine etrafında ve "x" eksenine boyunca olmaktadır. Bu bilgilere göre:

α : Servoya bağlı kolun yatay eksene yaptığı açı

β : Servoya bağlı kolun X eksenine yaptığı açı



Şekil 3: Motor hareket eksenleri

Yukarıdaki bilgilere göre platforma bağlı olan kol ve rotların konum vektörleri aksenel olarak hesaplanacaktır. Motorların dönüş yönleri birbirlerine zıt olduğu için çift sayılı motorların dönüşlerini α açısı kadar hesaplarken ters yönde dönen tek sayılı motorlar için $\pi - \alpha$ kadar döndüğünü varsayılacaktır. Aynı zamanda sabit kısım ile bağlantı yapılan konum merkezden x_{bi} kadar ötelenmiş olması hesaba katılmalıdır.

Dolayısıyla Çift sayılı motorların kol ve rot birleşim noktaları:

$$a_{xi} = a * \cos\alpha * \cos\beta + b_{xi} \quad (20)$$

$$a_{yi} = a * \cos\alpha * \sin\beta + b_{yi} \quad (21)$$

$$a_{zi} = a * \sin\alpha + b_{zi} \quad (22)$$

Tek sayılı motorlar için birleşim noktaları ise aşağıdaki gibi olur:

$$x_{ai} = a * \cos(\pi - \alpha) * \cos(\pi + \beta) + x_{bi} \quad (23)$$

$$y_{ai} = a * \cos(\pi - \alpha) * \sin(\pi + \beta) + y_{bi} \quad (24)$$

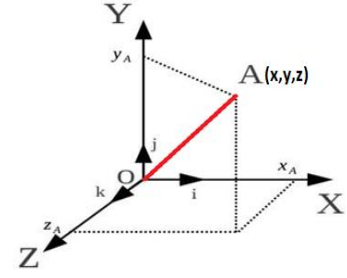
$$z_{ai} = a * \sin(\pi - \alpha) + z_{bi} \quad (25)$$

Ayrıca aşağıdaki trigonometrik dönüşüm eşitlikleri tek sayılı motorlar için yazılan eşitliklerde yerlerine yazıldığında çift sayılı motorların konum formülleri elde edilmiş olunacaktır.

$$\sin(\pi - \alpha) = \sin\alpha \text{ ve } \cos(\pi - \alpha) = -\cos\alpha \quad (26)$$

$$\sin(\pi + \beta) = -\sin\beta \text{ ve } \cos(\pi + \beta) = -\cos\beta \quad (27)$$

Pisagor teoremine göre üç boyutlu bir uzaydaki noktanın konumu aşağıdaki gibi ifade edilir [18].



Şekil 4: Pisagor teoremine göre 3 boyutlu hareket düzlemi

$$(A)^2 = (A_x - O_x)^2 + (A_y - O_y)^2 + (A_z - O_z)^2 \quad (28)$$

Dolayısıyla bu tasarımdaki hareketli platform konum hesaplamaları aşağıdaki gibi olacak.

$$(a_i)^2 = (a_{xi} - b_{xi})^2 + (a_{yi} - b_{yi})^2 + (a_{zi} - b_{zi})^2 \quad (29)$$

$$(l_i)^2 = (p_{xi} - b_{xi})^2 + (p_{yi} - b_{yi})^2 + (p_{zi} - b_{zi})^2 \quad (30)$$

$$(r_i)^2 = (p_{xi} - a_{xi})^2 + (p_{yi} - a_{yi})^2 + (p_{zi} - a_{zi})^2 \quad (31)$$

$(l_i)^2$ ve $(a_i)^2$ denklemlerini sadeleştirmek için $(r_i)^2$ denkleminde yerlerine yazıldığında aşağıdaki denklem elde edilir.

$$l_i^2 + a_i^2 - r_i^2 = 2 * (b_{xi}^2 + b_{yi}^2 + b_{zi}^2) + 2 * a_{xi} * (p_{xi} - b_{xi}) + 2 * a_{yi} * (p_{yi} - b_{yi}) + 2 * a_{zi} * (p_{zi} - b_{zi}) - 2 * (p_{xi} * b_{xi} + p_{yi} * b_{yi} + p_{zi} * b_{zi}) \quad (32)$$

Yukarıdaki denklemde a_{xi} , a_{yi} ve a_{zi} yerlerine yazıp parantezler dağıtılsa aşağıdaki denklem elde edilecektir.

$$l_i^2 + a_i^2 - r_i^2 = 2 * a * \sin\alpha * (p_{zi} - b_{zi}) + 2 * a * \cos\alpha * [\cos\beta * (p_{xi} - b_{xi}) + \sin\beta * (p_{yi} - b_{yi})] \quad (33)$$

Yukarıdaki denklem aşağıdaki denklem formatına gelmiş durumda [19]:

$$L = m * \sin\alpha + n * \cos\alpha = \sqrt{m^2 + n^2} * \sin(\alpha + p) \quad (34)$$

$$\tan p = \frac{n}{m} \text{ ve } p = -\text{Arctan}\left(-\frac{n}{m}\right) \quad (35)$$

$$L = \sqrt{m^2 + n^2} * \sin\left(\alpha + \text{Arctan}\left(\frac{n}{m}\right)\right) \quad (36)$$

$$\frac{L}{\sqrt{m^2 + n^2}} = \sin\left(\alpha + \text{Arctan}\left(\frac{n}{m}\right)\right) \quad (37)$$

$$\alpha = \text{Arcsin}\left(\frac{L}{\sqrt{m^2 + n^2}}\right) - \text{Arctan}\left(\frac{n}{m}\right) \quad (38)$$

Yukarıdaki denklemlere göre ilgili değerler aşağıdadır.

$$L = l_i^2 + a_i^2 - r_i^2 \quad (39)$$

$$m = 2 * a * (p_{zi} - b_{zi}) \quad (40)$$

$$n = 2 * a * [\cos\beta * (p_{xi} - b_{xi}) + \sin\beta * (p_{yi} - b_{yi})] \quad (41)$$

Yukarıdaki hesaplamalar ile belirli açılarla konumlandırılmış olan servoların ve bu servolara bağlanan kol ve rot uzunluklarını kullanarak servolardan hareketli platformun bağlantı noktalarına olan uzaklıkları hesaplanabilir. Fakat bu işlemden önce hareketli platformunun bazı değişmezlerinin belirlenmesi gerekmektedir. Bu değişmez değerler, platformun başlangıç durumu olacak ve hesaplamalar bu değerler ile yapılacaktır.

Hareket platformunun başlangıç durumu sabit ve hareketli platform kısmı birbirlerine paralel durumda olacak ve motorların kol kısımları birbirlerine bakacak şekilde konumlandırılmıştır. Bunun için motorların dönüş dereceleri 90 derece olarak belirlenecektir. Bu durumdayken platformda herhangi bir dönüş hareketi yapılmamış olacaktır. Bu durumda sabit ve platform kısımlarının merkezleri arasındaki değer başlangıç yükseklik değeri h_0 olarak adlandırılacaktır. Bu yükseklik değeri aynı zamanda T vektöründeki z eksenindeki konumunu belirtmiş olacaktır.

Dolayısıyla başlangıçta aşağıdaki durum elde edilir:

$$q_0 = T + R_{PB} * p_i \quad (42)$$

$$q_0 = \begin{bmatrix} 0 \\ 0 \\ h_0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (43)$$

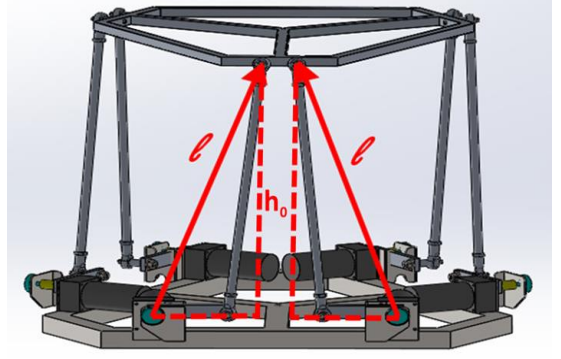
$$q_0 = \begin{bmatrix} p_x \\ p_y \\ h_0 + p_z \end{bmatrix} \quad (44)$$

$$(l_i)^2 = (p_{xi} - b_{xi})^2 + (p_{yi} - b_{yi})^2 + (h_0 + p_{zi} - b_{zi})^2 \quad (45)$$

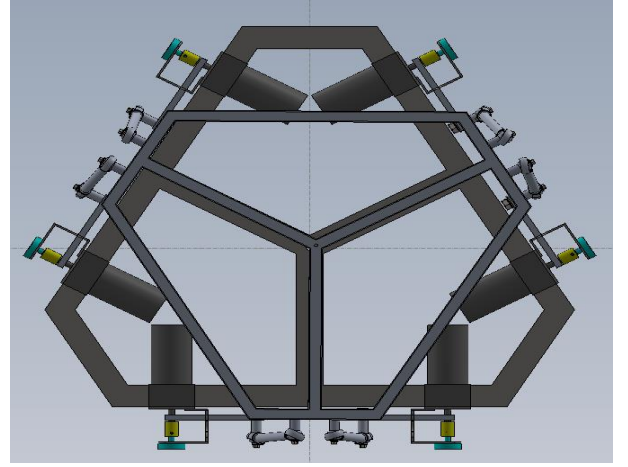
Eğer başlangıç pozisyonu olarak ilgili manipülatör 90 derece açıdayken, platforma bağlı olan rot tam olarak kol ile 90 dereceyken motor-platform vektör uzunluğu için aşağıdaki eşitlik elde edilir.

$$l_i^2 = r_i^2 + a_i^2 \quad (46)$$

Fakat bu tasarımda başlangıç durumu aşağıdaki görseldeki gibi olacaktır ve bu durumda yukarıdaki eşitlik olduğu gibi tasarım üzerinde uygulanamayacaktır. Platform hareketi ile kol ve rot arasındaki açı sürekli değişken olacaktır. Dolayısıyla diğer motorları da hesaba katarak yapılan bir hesaplama mikroişlemci üzerinde çok fazla işlem yüküne sebep olacaktır. İşlemin basitleştirilmesi adına bağlantı noktaları ve başlangıç nokta açıları ile aşağıdaki görselde kesik çizgiler ile gösterilen uzunluklar hesaplanacak.



Şekil 5: Başlangıç durumundaki hareketli platform yüksekliği



Şekil 6: Platformun üstten görünümü

Tasarımımıza tepeden bakıldığında rot ve kol parçalarının aynı hizada olduğu görülmektedir. Dolayısıyla başlangıçtaki yükseklik değeri hesaplanırken Y eksenini yok sayarak 2 boyutlu hesaplama yapılabilecektir. Aşağıdaki eşitlik herhangi bir motor üzerinden başlangıç yükseklik değerinin hesaplanmasını sağlayacaktır.

$$h_0 = \sqrt{(p_{xi} - b_{xi})^2 - l_0^2} \quad (47)$$

Ayrıca bu tasarımda kullanılan motorların istenilen dönüş ve kayma hareketleri için gelmesi gereken açının hesaplanması için yukarıdaki eşitlikler kullanılır. Bu hesaplama için ihtiyaç olan kinematik hesaplamalardan alınan her bir leg vektörü gerekmektedir. Ayrıca sabit kısmın yerde olduğu göz önünde bulundurulduğu için $b_z = 0$ olacaktır.

$$l_i = T + R_{PB} * p_i - b_i \quad (48)$$

$$l_0 = \begin{bmatrix} 0 \\ 0 \\ h_0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} - \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (49)$$

$$l_0 = \begin{bmatrix} p_x \\ p_y \\ h_0 + p_z \end{bmatrix} - \begin{bmatrix} b_x \\ b_y \\ 0 \end{bmatrix} \quad (50)$$

Yukarıdaki denklem sadeleştirildiğinde eşitlik 51 elde edilir. Yukarıdaki işlemler vektörel olarak yapıldığı için leg vektörünün uzunluğunu bulabilmek için Pisagor teoremi uygulandığında aşağıdaki sonuç elde edilecektir.

$$(l_i)^2 = (p_{xi} - b_{xi})^2 + (p_{yi} - b_{yi})^2 + (h_0 + p_{zi})^2 \quad (51)$$

Yukarıdaki eşitlikler aracılığıyla hareketli platformun hedef değerlerine ulaşabilmesi için gereken değerler hesaplanarak motorların bulunması gereken pozisyon açıları hesaplanmaktadır. Bu hesaplamalar sonraki bölümde sistem kontrolünü sağlayan programda kullanılacaktır.

4. Tasarlanan Modelin Prottip Devre ile Gerçeklenmesi

Bu kısım iki alt başlık halinde incelenecektir. İlk bölüm altında sistemin MS Flight Simulator uygulamasından anlık olarak uçuş simülasyon verilerinin alınıp elektronik sistem ile haberleştirilmesi gereken programlama işlemleri olacaktır. İkinci bölüm ise hareket platformunun kontrolünü sağlayan elektronik devrenin tasarımı üzerine olacaktır.

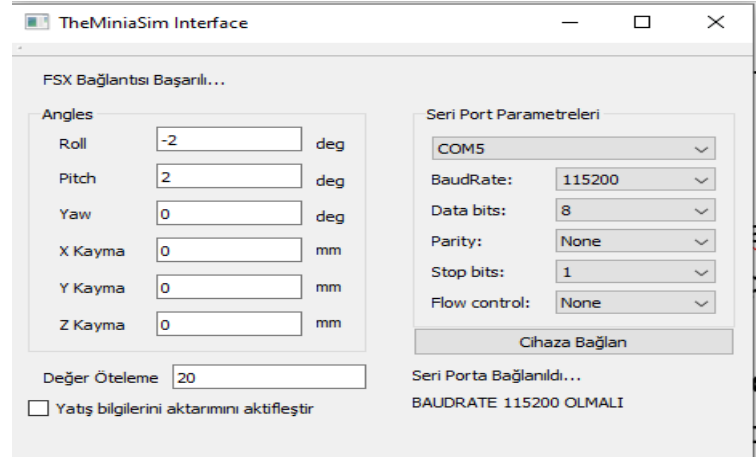
4.1 Programlama İşlemleri

Bu bölümde platformun hareketi için gerekli programlar yazılacaktır. Sistem yazılımları iki ana birimden oluşmaktadır. Bu birimlerden birisi MS Flight Simulator uygulamasından uçağın anlık verilerini okuyacak ve okunan bu değerleri seri port aracılığı ile motor hareketlerinden sorumlu birime gönderecektir. Bu ara yüz programı bilgisayar üzerinde çalışacaktır. Diğer birim ise seri porttan gelen uçuş bilgilerini işleyerek motor açılarını hesaplayacak ve motorları gerekli açılara getirip, bu açılarda tutmaktan sorumlu olacaktır.

Bilgisayar tarafında Qt Framework kullanılarak C++ dilinde geliştirilen bir ara yüz programı bulunmaktadır. Bu program ilk olarak Microsoft Flight Simulatör uygulamasına bağlanacaktır. Uygulamada simüle edilen uçuşun anlık olarak uçuş bilgileri alınacaktır. Alınan bu uçuş bilgileri seri port

üzerinden hareket sistemine gönderilecektir. Hareket sistemi 48 MHz hızında çalışan STM32F051 mikroişlemcisi ile yönetilmektedir. Bu işlemcinin hızı gerekli kinematik hesaplamalar için yeterli olduğu için hesaplamalar bu tarafta yapılacaktır. İşlemci tarafında ise seri port üzerinden UART DMA katmanı aracılığı ile veriler yatış ve dönüş verileri alınacaktır. Gelen veriler ile MicroVision Keil programı kullanılarak C dilinde yazılan program aracılığı ile 3. bölümde açıklanan kinematik hesaplamalar yapılacak. Hesaplamalar sonucunda motor dönüş açılarına göre PWM sinyali oluşturulup motorların ilgili açılara gelmesi ve bu açılar değişmedikçe motorların hesaplanan açılarda tutulması sağlanacaktır. Şekil 7'de tasarlanan ara yüz programından bir görsel bulunmaktadır

Şekil 7'den görüldüğü üzere ara yüz programı açıldığında Microsoft Flight Simulator X uygulamasına bağlanacak ve bu bilgiler kullanıcıya verilecektir. Bu işlemden sonra simüle edilen uçağın uçuş bilgileri anlık olarak alınmaktadır. Aynı zamanda Yatış bilgilerini aktarımı aktifleştir kutusu işaretli olduğunda bu veriler seri port aracılığı ile bağlı olunan seri port üzerinden gönderilecektir.

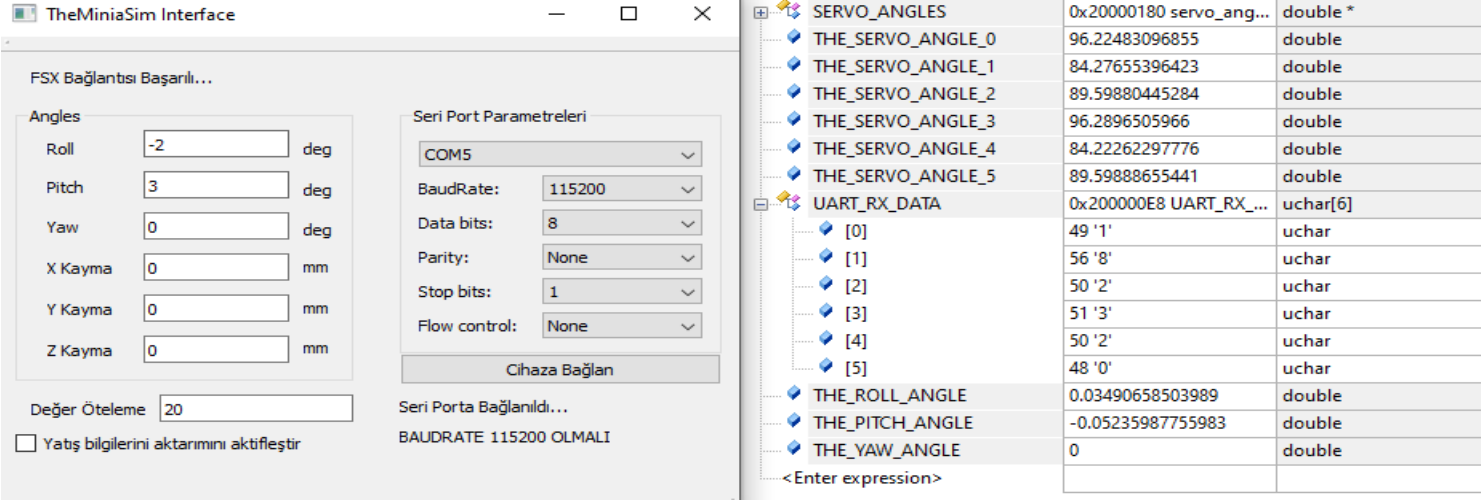


Şekil 7: TheMiniaSim Interface programı ekran görüntüsü

Burada dikkat edilmesi gereken durumlardan bir tanesi, gömülü sistem tarafında seri port haberleşme hızı 115200 Baud/s olarak ayarlanmış olmasıdır. Bu yüzden bizde programımızda seri port haberleşme yapılandırmasında BaudRate değerini aynı şekilde ayarlanması gerekmektedir. Aksi takdirde farklı değerler

gidecektir ve sistem hatalı çalışacaktır.

Şekil 8'de ara yüz programı ve mikroişlemcinin haberleşmesini ve mikroişlemcinin hata ayıklama ekranından alınan veriler görüntülenmektedir.



Şekil 8: TheMiniaSim Interface arayüz programı ve Mikroişlemci Debug ekranı

Şekil 8'de görüldüğü gibi değerler ötelenmiş olarak doğru bir şekilde gönderilmekte ve kinematik hesaplamalardan sonra servo motor açıları hesaplanmaktadır. ROLL ve PITCH değerlerinden görüldüğü üzere açılar radyan olarak hesaplanmaktadır. Programlama dillerinde açısal hesaplamalar radyan değerler üzerinden yapılmaktadır. Bu yüzden işleme başlamadan önce ilgili değerler radyana dönüştürüldü ve kinematik hesaplamaların yapılması sağlandı. Oluşturulan bir kütüphane ile ana fonksiyonundan sadece ilgili fonksiyonları çağrılarak gerekli hesaplama sonuçları bir dizi olarak döndürülmektedir. Dönen bu dizideki açısal değerler tek ve çift sırasına göre ilgili değişkenlerine atandıktan sonra PWM fonksiyonları çağrılmaktadır.

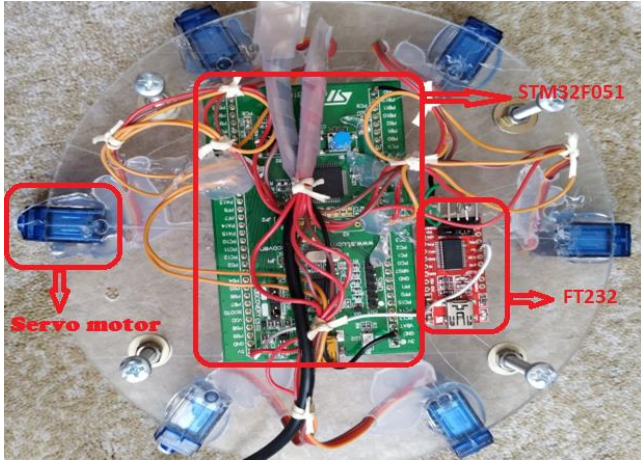
Seri port üzerinden mikroişlemciye gelen değerler karakter olarak alınıp işlendikten sonra arayüz tarafından yapılan öteleme geri alınıp ardından radyana dönüştürülmektedir. Bu işlemlerden sonra ilgili sırasına göre ilgili ROLL, PITCH ve YAW değişkenlerine atanmaktadır. Bu hesaplamaların sürekli yapılarak sistemin çok fazla enerji harcayarak mikroişlemcinin çok fazla ısınmasına yol açacaktır. Bu yüzden haberleşme katmanı fonksiyonu kesme olarak tanımlandığı için her seri port haberleşmesi her tamamlandığında çalışacaktır. Dolayısıyla çok fazla matematiksel işlem barındıran hesaplamaları burada yapmak iletişime zarar verecektir ve aksatacaktır. Dolayısıyla kinematik hesaplamalar ana fonksiyonun içinde gerçekleştirilmektedir.

Ana program döngüsü her hesaplamadan ardından 1milisaniye kadar bekleyecek ve UART DMA katmanını aktifleştirecek. Bu katman seri port haberleşmesinin asenkron bir şekilde yapılmasını ve ana program döngüsünde zaman kaybını engelleyecektir. Bu esnada seri port üzerinden veri gelme durumunu belirten değişkenin durumuna göre veri gelmemişse bir işlem yapılmayacaktır. Eğer veri gelmişse de hesaplamalar için gerekli fonksiyona bu verileri göndererek geriye açılardan olduğu bir dizi döndürülecektir. Gelen bu veriler PWM sinyali oluşturulmak üzere ilgili fonksiyona gönderilecektir.

PWM oluşturma için önceden hazırlanmış STM32 mimarisinde bulunan Timer donanımı kullanılmaktadır. Dahili Timer donanımı işlemcinin beslendiği dahili 48MHz frekansta clock sinyali üreten osilatörden beslendiği için burada oluşacak olan sinyalde aynı şekilde 48MHz olacaktır. Fakat bu tasarımda kullanılan servo motorlar, bu frekanstaki PWM sinyalini algılayamayacağı için bu sinyalin belirli değerlerdeki ön bölücüler ile belünmesi gerekmekte ve uygun değere ayarlanması gerekmektedir. Ardından bu değerler ile doğru bir şekilde PWM sinyali oluşturmak için yine STM32 standart kütüphanelerinde gelen PWM sinyali oluşturma yapısı ile sinyal oluşturulacak ve ayarlanan PWM sinyalinde tutulacaktır. STM32 standart kütüphanesi yardımıyla Timer donanım katmanı oluşturulan PWM sinyalini otomatik olarak tutmaktadır.

4.2 Elektronik Devre Tasarımı

Bu çalışma için geliştirilen TheMiniaSim uygulaması Qt Framework kullanılarak C++ dilinde geliştirilmiştir. Bu çalışma için kullanılan elektronik kartlar Şekil 9'da gösterilmiştir. Bu program ile simüle edilen uçuşun verileri, gerekli kinematik hesaplamaların yapılarak servo motorlara aktarıldığı ARM Cortex M0 tabanlı işlemciye FT232 Seri port haberleşme çipi aracılığı ile gönderilmektedir. FT232 ve STM32F0 arasındaki haberleşme Universal Aseynchron Receive-Transmit(UART) protokolü kullanılmaktadır. Aynı zamanda UART protokolü ile birlikte ARM mimarisine sahip işlemcilerde bulunan DMA katmanı birlikte çalışmaktadır. DMA katmanı sayesinde UART protokolünden gelen veriler programın normal çalışmasını engellemeden eş zamanlı olarak alınabilmektedir.



Şekil 9: Prototip devre tasarımı ve konumlandırılması

Mikroişlemci düşük fiyat mantığında üretilen bir işlemci olmasına rağmen kendi içerisinde 48MHz frekansında clock sinyali üretebilmektedir. Bu frekans değeri sayesinde işlemcinin işlem hızını ciddi oranda yükseltmektedir ve kinematik hesaplamalar için yeterli bir konuma gelmektedir. Kinematik hesaplamalardan sonra oluşan açısal değerleri Timer donanım katmanı aracılığıyla PWM sinyali oluşturarak SG90 servo motorlara aktarır PWM sinyalini sürekli belirlenen değerde tutarak, motorların dışardan bir kuvvet gelse bile aynı açıda kalmalarını sağlamaktadır.

Bu prototip için seçilen servo motorun özelliği, gönderilen PWM değerlerine göre açısal değerlere gelir ve orada kalır [20]. Motorların konum bilgisini almak için ek olarak harici bir sensöre gerek yoktur. Aynı zamanda bütün sistem gibi 5V gerilimde çalışması ve bu gerilimde normal çalışma esnasında 800g-1000g gibi boyutuna göre yüksek ağırlık altındada çalışabilmesi bu motoru avantajlı kılmaktadır.

Platform tasarımı için en önemli konu ise kinematik hesaplamalar için gerekli olan bağlantıların konumlarıdır. Bu konular 3 boyutlu eksenel sistemde kesin olarak bilinmelidir. Bu sayede matematiksel hesaplamaları vektörel olarak doğru bir şekilde yapabilmektedir. Platformun sabit ve hareketli kısımları daire şeklinde kesilmiştir. Kinematik hesaplamalar için base-arm, arm-rot ve rot-platform bağlantı noktalarının bilinmesi gerekmektedir. Bu hesaplamalar program çalışmaya başlamadan önce yapılmalı ve sabit değerler olarak programın normal hesaplamalarında kullanılmalıdır. Tasarımda kullanılan arm yapılarının uzunluğu 15mm ve rotların uzunluğu 95 mm olarak ayarlanmıştır. Sabit kısımda monte edilen herbir motorun konumlarının hesaplanması için motorlar çiftler olarak alınmıştır ve motor çiftlerindeki açılar 60 derece olarak belirlenmiştir. Motorlar çiftleri de sabit platforma 120 derecelik açılar ile sabitlenmiştir. Sabit kısmın merkezinden motor hareketinin sağlandığı motor-arm noktası arası mesafe ise 102mm olarak belirlenmiştir. Açısal değerler hareketli platform içinde aynı şekilde tasarlanmış olur dairesel hareketli platformunda merkezinden rot ile olan bağlantı noktalarına 67mm olarak belirlenmiştir. Başlangıç durumdaki yükseklik 88mm olarak belirlenmiştir. Bütün bu değerler ile yapılan hesaplamalar, kinematik hesaplamalar yapılmaya başlanmadan önce yapıp bağlantı noktalarının konumlarını hesaplayarak programda kullanılmak üzere işlemci programındaki ilgili değişkenlere atanmalıdır.

4.3 Çalışmanın Maliyet Analizi

Bu çalışma düşük bütçeli bir tasarım olduğu için fiyat-performans açısından kullanılan işlemcinin tepki süresi önemli bir etkidir. Bu yüzden seçilen işlemcinin de piyasadaki en performanslı ve en ucuz işlemci olmasına dikkat edildi. Bu yüzden ARM Cortex M-0 mimarisi üzerine kurulu olan STM32F051 mikroişlemcisi seçilmiştir. Bununla birlikte hareketi esnasında konum bilgisini alabilmek için sensör gerektirmeyen motorlar seçilmiştir. Geri kalan parçalar modelde belirlendiği şekilde kolayca üretilmiş ve sadece malzeme maliyeti dikkate alınmıştır. Bütün bu malzemelerin maliyeti göz önüne alındığında toplam maliyet 25USD altında olmaktadır. Aşağıdaki tabloda ürünlerin maliyetleri verilmiştir.

Tablo 1: Parça Maliyet listesi

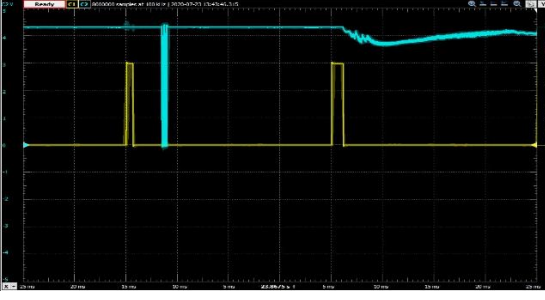
Parça Adı	Maliyeti
STM32F051 Development Board	9,90 USD
SG90 Servo Motor x6	6,90 USD
FT232 Serial Port Device	2.30 USD
Diğer malzemeler	6,00 USD

5. Çalışmanın Sonuçları ve Öneriler

Bütün motorların dönüş açılarının bulunabilmesi için yukarıdaki belirtilen hesaplamaların çok hızlı bir şekilde gerçekleştirilmesi gerekmektedir. Aşağıdaki osiloskop çıktısında eş zamanlı olarak alınan seri port sinyali ile PWM sinyali görülmektedir. Sarı renk ile gösterilen sinyal STM32F051 mikroişlemcimizin TIMER donanımından gelen PWM sinyalidir. Aynı zamanda mavi renk ile gösterilen sinyal ise seri port üzerinden gelen haberleşme sinyalidir.

Şekil 10'dan görüldüğü üzere haberleşme sinyalinden önceki PWM sinyali 700us olarak gelmektedir. Haberleşme sinyalinden sonraki PWM sinyali ise 1200us olarak devam etmektedir. Bu değer değişim grafiği sadece bir motor için gösterilmiştir. Mikro işlemci programı tasarımına göre çıktılar bütün hesaplamalardan sonra yapılmaktadır. Arka arkaya gelen iki PWM sinyalinin arasında 20ms süre bulunmaktadır. Görülmektedir ki 6 motor için hesaplamaların yapılması ve motorlara aktarılması 20ms'den daha kısa bir süre almaktadır. Bu grafikte hız konusunda seçilen işlemcinin yeterli olduğunu göstermektedir.

Seçilen işlemcimizin donanım katmanları ile gereken işlemleri yapması ayrıca tasarımın faydasına olmaktadır.

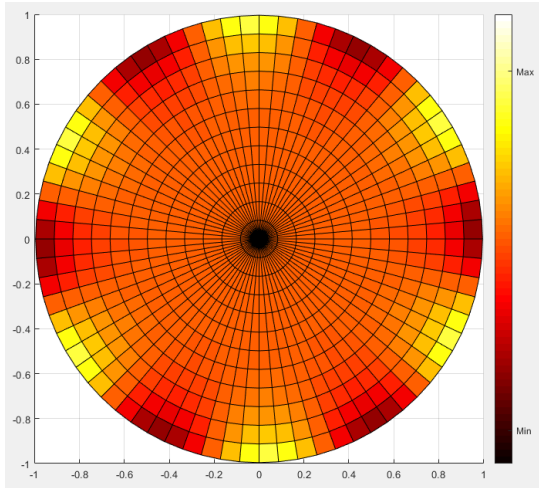


Şekil 10: PWM ve UART haberleşme sinyalleri Scope görüntüsü

Tablo 1. Çalışmada Kullanılan Bakteriler ve İnkübasyon Koşulları

Roll	Pitch	Yaw	Servo0	Servo1	Servo2	Servo3	Servo4	Servo5
0	0	0	90	90	90	90	90	90
10	0	0	72	90	106	106	90	72
20	0	0	53	90	125	125	90	53
10	10	0	82	109	115	97	70	63
0	20	10	104	138	103	77	44	81
-10	0	-10	113	85	82	68	98	101
0	-20	0	71	48	69	107	130	109
-20	0	-20	140	83	80	50	109	119
15	15	15	74	131	126	115	54	68
-15	-15	-15	109	54	59	74	131	126

Tablo 1’de bulunan açısal değerler ışığında prototip Stewart platformun hareket kabiliyeti ± 150 olduğu görülmektedir. Şekil-11 deki grafiğe göre bu hareket esnasında platform tasarımının hareketli kısmının incelendiğinde platformun bağlantı noktalarının merkeze göre daha fazla hareket ettiği görülmektedir. Bağlantı noktaları arasında kalan bölgelerin ise hareket miktarının daha az olduğunu görülecektir.



Şekil 11: Platform Hareket kabiliyetini gösteren MATLAB diyagramı

Şekil 11 Matlab programında hazırlanmıştır. Bu diyagrama göre Stewart platformun hareketli kısmın hareket edebilme kapasiteleri gösterilmiştir. Dönüş hareketleri merkezden yapıldığı için merkez noktası sabit olarak görülmektedir. Diyagramın bağlantı noktaları küçük hareketler ederken bağlantı noktaları arası daha fazla hareket etmektedir. Uç kısımlardaki küçük kırmızı bölgeler platform bağlantı

Kinematik hesaplamalar sonucunda her bir motor için oluşan açısal değerler tablosu aşağıdaki gibi olacaktır. Stewart platformu hesaplamalarını mikro işlemci önceki bölümlerde belirtilen matematiksel modele göre yapılacak ve PWM çıkışlarından ilgili açığa göre PWM çıktısı oluşturulacaktır.

bölgeleridir. Bu bölgeler etrafındaki dönüşler bağlantı noktaları etrafında olduğu için diğer bölgelere göre hareket kabiliyeti daha azdır fakat merkez noktalar gibi en az değildir.

Şekil-12’de ise prototip olarak tasarlanan uçuş simülörünün gerçek görüntüsü bulunmaktadır.



Şekil 12: Küçük Boyutlu 6DOF Prototip Uçuş Platformu Tasarımı Gerçek Görüntüsü

Stewart platformun hareketi esnasında motorlara dönüş açıları gönderildiğinde motorlar direk olarak o açılara gitmektedirler. Bu hareketlerini herhangi bir kontrol olmadan en hızlı şekilde yapmaktadırlar. Bu sebeple sistemin hareket platformunun tepkileri çok sert olmaktadır. Sistemin sert tepkileri ile sert dönüşler yapılmaktadır. Sert dönüşler tasarımın mekanikğine zarar verebilmektedir. Ayrıca daha büyük tasarımlarda ani yüklenmelere neden olabilmektedir. Bu yüzden sistem tepki süresinden taviz vermeden daha yumuşak bir şekilde hareket etmelidir. Bu özellik hareket sistemine PID kontrol mekanizması eklenmesiyle sağlanabilir. PID ile sistem hızlı bir şekilde yumuşak bir şekilde hareket edecektir ve sistemin ani yüklenmelerinin de önüne geçecektir.

Aynı zamanda bu tasarımda kullanılan servo motorlar direk olarak gönderilen PWM açılara gitmektedirler. Herhangi bir

kontrol mekanizmasına sahip değildir. Dolayısıyla işlemci sadece komutu gönderir ve hedef açığa gelmesiyle ilgilenmez. Sistemin doğru çalışması için motorlarında ilgili konumlara gelmesi gerekmektedir. Bu sebeple bu açıların bir sensör yardımıyla kontrol ediliyor olması gerekmektedir. Özellikle daha büyük tasarımlarda mekanik bağlantılara zarar verilmesini önlemek adına sistemin doğru çalıştığını kontrol etmek gerekmektedir. Bununla birlikte sisteme entegre edilmesi önerilen PID sistemi de motorların anlık açılarına ihtiyaç duymaktadır. Bu nedenle ilgili sensörlerden geri besleme sinyali ile ilgili PID girişleri beslenmelidir.

5. Teşekkür

Proje desteklerinden dolayı Kocaeli Üniversitesi BAP koordinasyon birimine ve Fen Bilimleri Enstitüsü Müdürlüğüne teşekkür ederiz.

Simgeler

X, Y ve Z : Bir noktanın döndürülmeden önce eksenlerinin

X : Bir noktanın merkez ile x eksenindeki açı

Y : Bir noktanın merkez ile y eksenindeki açı

Z : Bir noktanın merkez ile z eksenindeki açı

Kaynakça

- [1] D. Stewart, "A platform with six degrees of freedom," Proc. Inst. Mech. Engr., vol. 180(1), pp. 371–386, 1965.
- [2] Realization of a Desktop Flight Simulation System for Motion-cueing Studies, Berkay Volkaner, S. Numan Sozen1 and V. Emre Omurlu, DOI: 10.5772/63239
- [3] Development of a motion simulator for testing a mobile surveillance robot, Journal of Mechanical Science and Technology 23 (2009) 1065-1070
- [4] Filip Szufnarowski "Stewart platform with fixed rotary actuators: a low cost design study"
- [5] Transfer of Training from a Full-Flight Simulator Vs. a High-Level Flight-Training Device with a Dynamic Seat, August 2010, DOI: 10.2514/6.2010-8218
- [6] de Winter, J. C. F., Dodou, D., Mulder, M., 2012. Training effectiveness of whole body flight simulator motion: A comprehensive meta-analysis. The International Journal of Aviation Psychology 22(2), 164–183.
- [7] Rasmussen, J., 1983. Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models. IEEE Transactions on Systems, Man, and Cybernetics SMC-13(3), 257–266.
- [8] Mulder, M., van Paassen, M. M., Boer, E. R., 2004. Exploring the Roles of Information in the Control of Vehicular Locomotion: From Kinematics and Dynamics to Cybernetics. Presence: Teleoperators and Virtual Environments 13(5), 535–548.
- [9] Stapleford, R. L., Peters, R. A., Alex, F. R., 1969. Experiments and a Model for Pilot Dynamics with Visual and Motion Inputs. Tech. Rep. NASA CR-1325, NASA.
- [10] Zaal, P. M. T., Pool, D. M., Chu, Q. P., van Paassen, M. M., Mulder, M., Mulder, J. A., 2009a. Modeling Human Multimodal Perception and Control Using Genetic Maximum Likelihood Estimation. Journal of Guidance, Control, and Dynamics 32(4), 1089–1099.
- [11] Application of H1 Theory to a 6 DOF Flight Simulator Motion Base, Mauricio Becerra-Vargas and Eduardo

ω, γ ve φ açıları : Sırasıyla Roll Pitch ve Yaw açıları

Cos : Cosinus

Sin : Sinus

x' : x eksenindeki dönüşten sonra x eksenine olan uzaklık

y' : y eksenindeki dönüşten sonra y eksenine olan uzaklık

z' : z eksenindeki dönüşten sonra z eksenine olan uzaklık

d_x : Bir noktanın x eksenine olan uzaklığı

d_y : Bir noktanın y eksenine olan uzaklığı

d_z : Bir noktanın z eksenine olan uzaklığı

R: 3 eksen etrafında dönüş

R_x : x eksenindeki dönüş

R_y : y eksenindeki dönüş

R_z : z eksenindeki dönüş

P_i ve p_i

: Platformun Rod ile bağlandığı noktalar ve vektörleri

B_i ve b_i

: Servoların Base kısma bağlandığı noktalar ve vektörleri

A_i ve a_i : Servo kolu ile rodun birleştirildiği noktalar ve vektörleri

r : Rod uzunluğu

l_i : B_i ile P_i noktaları arasındaki vektör

h_0 : Başlangıç durumunda hareketli platformun yüksekliği

- [12] Morgado Belo, University of Sao Paulo – EESC, Department of Aeronautical Engineering
- [12] Salcudean, S., P. Drexel, D. Ben-Dov, A. Taylor, and P. Lawrence (1994). A six degree-of-freedom, hydraulic, one person motion simulator. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego-CA, pp. 2437-2443. IEEE.
- [13] Graf, R., R. Vierling, and R. Dillman (1998). A exible controller for a Stewart platform. In Proceedings of the 1998 Second International Conference on Knowledge-Based Intelligent Electronic Systems, Adelaide-SAAustralia, pp. 52-59. IEEE.
- [14] Sciavicco, L. and B. Siciliano (2005). Modeling and Control of Robot Manipulators (2 ed.). Londres: Springer-Verlag.
- [15] An Introduction to Mathematical Modelling, Glenn Marion, Bioinformatics and Statistics Scotland Given 2008 by Daniel Lawson and Glenn Marion
- [16] Equations of Motion of a Spin-Stabilized Projectile for Flight Stability Testing; Journal of Theoretical and Applied Mechanics 51, 1, pp. 235-246, Warsaw 2013
- [17] <https://mememememememe.me/post/stewart-platform-math/> [Erişim Tarihi: 11-Ağustos-2020]
- [18] Geometry of the 3D Pythagoras' Theorem, Journal of Mathematics Research; Vol. 8, No. 6; December 2016 ISSN 1916-9795 E-ISSN 1916-9809
- [19] <http://thetrig.blogspot.com/2011/11/m-sin-x-n-cos-x.html> [Erişim Tarihi: 11-Ağustos-2020]
- [20] http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg_90_datasheet.pdf [Erişim Tarihi: 12-Ağustos-2020]