

TEKNOFEST 2019
SÜRÜ İHA SİMÜLASYON YARIŞMASI
TEKNİK ŞARTNAMESİ

İÇERİK

1	KAPSAM	3
2	TANIMLAR	3
3	DEĞİŞKENLER.....	4
4	GENEL BİLGİLER	5
5	EN İYİLEME PROBLEMİNİN KAPSAMI	5
6	SİSTEM KURULUMU.....	6
7	YAZILIM GELİŞTİRME KİTİ (YGK) MODÜLLERİ	9
8	YARIŞMA KURALLARI	18
9	EK-1	20

1 KAPSAM

Bu belge Teknofest İstanbul web sayfasında yayında olan "Yarışma Şartnamesi" belgesinde^[1] detayları açıklanan yarışmanın teknik konularını detaylı açıklayan teknik şartnamedir. Yarışmacıların bu belgeyi okumadan önce "Yarışma Şartnamesi"ni okumuş olmaları gerekmektedir. Yarışmacıların sürü zekası/mühendisliği alanında literatür taraması yaptıktan sonra tasarım ve geliştirme faaliyetlerine başlamaları tavsiye edilmektedir. Sürü sistemleri hakkında örnek akademik literatür^{[2],[3]} incelemelerinize sunulmuştur.

[1] https://www.teknofestistanbul.org/Content/files/2019_satnameler/suru-ih-simulasyon-yarismasi_08.pdf

[2] M. Brambilla, E. Ferrante, M. Birattari ve M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective", Swarm Intelligence, cilt 7.1, pp. 1-41, 2013.

[3] L. Bayındır, "A review of swarm robotics tasks", Neurocomputing, no. 176, pp. 292-321, 2016.

2 TANIMLAR

Docker^[6]: İşletim sistemi seviyesinde sanallaştırma sağlayan konteyner yazılımı

Gazebo^[4]: Açık kaynaklı 3 boyutlu robotik simülatörü

İHA: İnsansız hava aracı (insansız helikopterler kullanılacaktır, helikopter modeli yarışma komitesi tarafından sağlanacaktır.)

Sahne: Gazebo görsel kullanıcı arayüzünde görünen sanal dünya

Sürü: Simülasyonda yarışmacının tüm İHA'larının bir araya gelerek oluşturduğu İHA sürüsü

Birey: Sürüde bulunan her bir İHA

Oturum: Yarışma komitesi tarafından simülasyonun sıfırdan koşturulması için yarışmacıya tanınan zaman

ROS^[5]: Robot operating system

THVKS: Telekom hizmeti verilen kazazede sayısı

Yarışma şartnamesi^[1]: Teknofest İstanbul web sayfasında yayında olan ve yarışmanın detaylarını anlatan belge

YGK: Yarışmacıların simülasyon sahnesindeki nesnelerin durumlarını sorgulayabilmeleri ve İHA'larına komut gönderebilmeleri için yarışma komitesi tarafından hazırlanmış yazılım geliştirme kiti

[4] <http://gazebosim.org>

[5] <http://www.ros.org>

[6] <https://www.docker.com>

3 DEĞİŞKENLER

YGK değişkenlerinin yayınlanacağı örnek veri paketi bu dokümanın sonundaki EK-1'de verilmiştir. Veri paketi "JSON" formatındadır [7].

active_uav: Betiğin kontrol ettiği İHA'nın YGK bilgi paketindeki etiketi

denied_zones: Uçuşa yasaklı bölgeler

fuel_reserve: Betiğin kontrol ettiği İHA'nın yakıt rezervi

level_difficulty: Oturum zorluk seviyesi

logical_camera_height_max: Sanal kameranın azami çalışma yüksekliği

logical_camera_height_min: Sanal kameranın asgari çalışma yüksekliği

logical_camera_fov: Sanal kameranın görüş açısı

hospital_quota: Hastanenin an itibariyle alabileceği ek yaralı sayısı

injured_pick_up_duration: Yaralıyı almak için İHA'nın havada asılı kalması gereken asgari süre

injured_pick_up_height: Yaralıyı almak için İHA'nın alçalması gereken asgari irtifa

injured_release_duration: Yaralı bırakma asgari süresi

injured_release_height: Yaralı bırakma asgari yüksekliği

round_score: Oturumdan alınan toplam puan

special_assets: Senaryoda konumu değişmeyen yapılar

uav_communication_range: Bir İHA'nın başka bir İHA ile ya da hastane ile iletişim kurabileceği azami menzil

uav_count: Sahnedeki İHA sayısı

telecom_served_people_count: Bir İHA'nın anlık olarak telekom hizmeti verdiği kazazede sayısı

telecom_radius: İHA'nın kapsama alanının yerdeki izdüşümü dairesinin azami yarıçapı (bknz. Şekil-3)

telecom_height_max: İHA'nın telekom hizmeti verebileceği azami yükseklik irtifa (bknz. Şekil-3)

telecom_height_min: İHA'nın telekom hizmeti verebileceği asgari yükseklik irtifa (bknz. Şekil-3)

world_width: Sahnenin eni

world_length: Sahnenin boyu

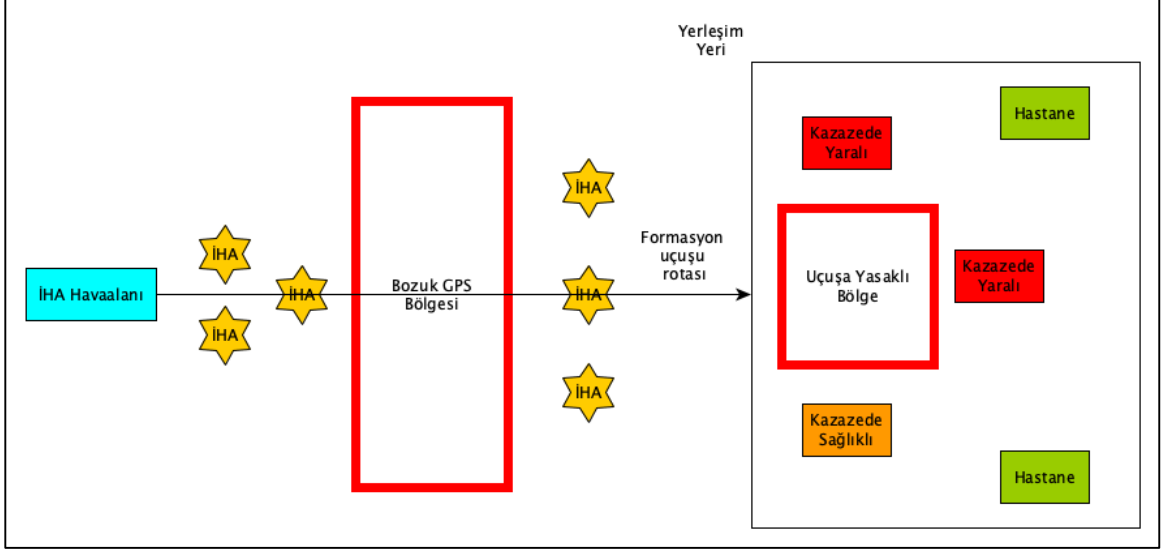
uav_guide: Kılavuz İHA

dispatch: Formasyon aşamasının bitiş sinyali

[7] <https://www.json.org/>

4 GENEL BİLGİLER

1. Simülasyon sahnesinde İHA, kazazede (insan), hastane, bina nesne sınıflarına ait nesnelere yer alacaktır. Sahnenin bina içermeyen ve nesne sayısı azaltılmış temsili krokisi Şekil-1'de verilmiştir.



Şekil-1: Simülasyon sahnesinin temsili krokisi

2. Kazazedeler sağlıklı ya da yaralı olmak üzere iki sınıfa ayrılmıştır.
3. Sürü tüm kazazedelere (yaralı veya sağlıklı) telekom hizmeti sağlayabilecektir.
4. Sürü sadece yaralı insanlara hastaneye taşıma hizmeti sunabilecektir.
5. Yarışma süresince senaryo zorluk seviyesine göre değişiklik gösterecek parametrelerin değerleri yarışmacılar tarafından simülasyon turunun başlangıcında YGK üzerinden okunabilecektir.

5 EN İYİLEME PROBLEMİNİN KAPSAMI

Yarışmacıların aşağıdaki kısıtlamalara ve amaçlara bağlı olarak ortaya çıkan eniyileme problemini, sürü sistemleri/zekası yaklaşımlarını (müşterek görev icrası, müşterek planlama, müşterek algılama, alan paylaşımı, kaynak paylaşımı vb.) kullanarak en yüksek puanı alacak şekilde çözmeleri beklenmektedir.

1. İHA'nın telekom hizmeti verebileceği insan sayısı kısıtlıdır.
2. İHA'nın telekom hizmeti verebileceği alan kısıtlıdır.
3. İHA'nın yaralıları tespit etmek için kullandığı kameranın görüş alanı kısıtlıdır.
4. İHA'nın bir kerede hastaneye taşıyabileceği yaralı sayısı kısıtlıdır.
5. İHA'ların iletişim donanımlarının menzilleri (uav_communication_range) kısıtlıdır. Bir İHA kendi iletişim menzilinin dışında kalan hastanelerin verilerine erişemez. Bir İHA kendi iletişim menzilinin dışında kalan diğer İHA'ların verilerine doğrudan erişemez fakat menzili içerisindeki diğer İHA'lar üzerinden erişebilir.
6. İHA'ların yakıtları kısıtlıdır. Yakıt tüketimini eniyileyecek şekilde uçuş rotaları belirlenmelidir.

6 SİSTEM KURULUMU

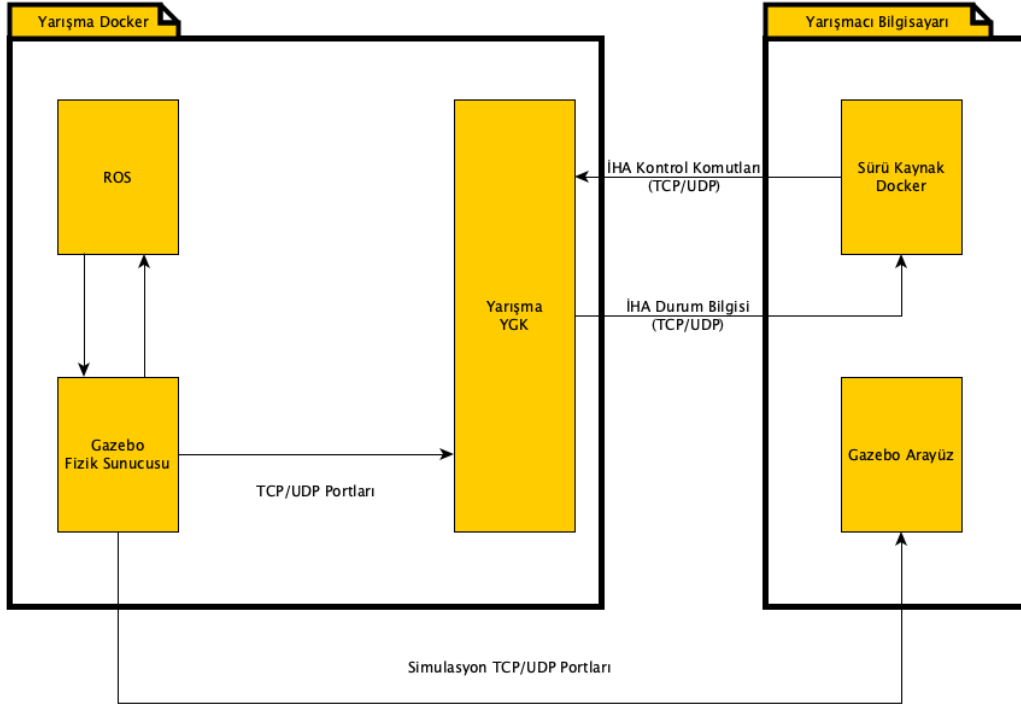
Minimum Sistem Gereksinimleri

1. CPU: Asgari Intel i5 ve denk olan işlemciler
2. RAM: Asgari 8GB
3. Harici GPU: Nvidia GPU'ları tavsiye edilir.
4. Saklama Alanı: Minimum 10GB boş disk alanı

İşletim Sistemi ve Programlar

1. Ubuntu 16.04 64-bit
2. ROS Kinetic
3. Gazebo 9
4. Yarışma YGK Deposu

Yarışma Simülasyonu Geliştirme Ortamı Genel Mimarisi



Şekil-2: Simülasyon ortamı genel mimarisi

Yarışmacılara kendi bilgisayarlarında geliştirme yapabilmelerine imkan sağlayacak "Yarışma Docker" konteyner imajı verilecektir. Bu konteyneri yarışmacılar kendi bilgisayarlarına indirip çalıştıracaklardır (bkz. Docker Kurulumu). Bu konteyner "Gazebo Sunucusu"nu ve "Yarışma YGK"sını içerecektir. Yarışmacılar kendilerine açılan TCP/UDP portlarına bağlanarak "Yarışma Docker" konteyneri ile iletişim kuracaklardır.

Yarışmacılar kendi bilgisayarlarına "Gazebo İstemcisi" programını kuracaklardır (bkz. Gazebo Kurulum) ve yarışma komitesi tarafından tanımlanacak TCP/UDP portları üzerinden "Gazebo İstemcisi"ni ROS'a bağlayacaklardır. Yarışmacılar kendi bilgisayarlarında "Sürü Kaynak Docker" konteynerini oluşturacaklardır ve içerisinde kendi sürü uygulamalarının betiklerini koşturacaklardır.

Yarısmaca takvimi uyarınca yarısmacılarla paylaşılacak olan “ Örnek Yarısmacı Kodu” ile yarısmacıların ortam (Yarısm Docker’ı) ile nasıl bağlantı kurabilecekleri gösterilecektir.

Docker CE Kurulumu

1. Ubuntu paket indexlerini güncelliyoruz.

```
$ sudo apt-get update
```

2. Docker repolarının HTTPS bağlantısı üzerinden yüklenebilmesi için gerekli olan paketleri yüklüyoruz.

```
$ sudo apt-get install \  
apt-transport-https \  
ca-certificates \  
curl \  
gnupg-agent \  
software-properties-common
```

3. Docker’ın resmi GPG anahtarlarını ekliyoruz.

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add - apt-transport-https \  
$ sudo apt-get update
```

4. Eklenen anahtarın doğruluğunu kontrol ediyoruz.

```
$ sudo apt-key fingerprint 0EBFCD88
```

5. Docker repolarını yüklüyoruz.

```
$ sudo add-apt-repository \  
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \  
$(lsb_release -cs) \  
stable"
```

6. Son yaptığımız index güncellemelerinin gerçekleşmesi için Ubuntu paket indexlerini tekrar güncelliyoruz.

```
$ sudo apt-get update
```

7. Docker paketlerini kuruyoruz.

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Gazebo9 Kurulumu

1. Gazebo depolarını Ubuntu'nun paket yöneticisine ekliyoruz.

```
$ sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu `lsb_release -cs` main" > /etc/apt/sources.list.d/gazebo-latest.list'
```

2. Depo anahtarlarını kuruyoruz.

```
$ wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -
```

3. Paket yöneticisini güncelliyoruz.

```
$ sudo apt-get update
```

4. Kullanılacak Gazebo paketlerini kuruyoruz.

```
$ sudo apt-get install gazebo9 libgazebo9-dev
```


7 YAZILIM GELİŞTİRME KİTİ (YGK) MODÜLLERİ

YGK şu modüllerden oluşacaktır:

1. Telekomünikasyon Hizmeti Modülü
2. Yaralı Tespit ve Nakil Modülü
3. İletişim Modülü
4. Yakıt Kontrol Modülü
5. Puan Hesaplama Modülü
6. Senaryo Kontrol Modülü
7. İHA Uçuş Kontrol Modülü
8. İHA Algılayıcı Modülü

Telekomünikasyon Hizmeti Modülü

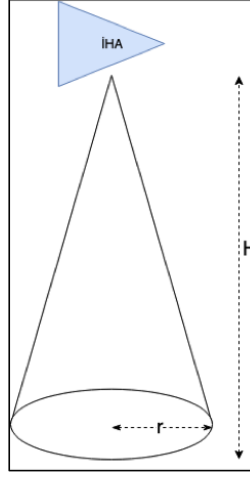
1. Bir İHA'nın üzerindeki baz istasyonunun kapsama alanı Şekil-3'deki gibidir. Yerdeki "r" yarıçaplı dairenin içerisindeki kazazedelerin telekom hizmeti alabileceği varsayılacaktır.
2. "r" yarıçapı YGK üzerinde **telecom_radius** değişkeni adıyla temsil edilecektir. Bu değişken sahne boyutlarına, hizmet veren İHA sayısına (**uav_count**) ve senaryonun zorluğuna (**level_difficulty**) göre yarışma sırasında değişiklik gösterecektir. "r" parametresi İHA'nın irtifasına göre değişmemektedir. "r"nin alabileceği azami değer aşağıdaki formüle göre hesaplanacaktır. (w: senaryo dünyasının eni (km), l: senaryo dünyasının boyu (km), n: sürüdeki İHA sayısı)

$$0 < r < \sqrt{\frac{w \cdot l}{n\pi}}$$

3. Sürü ne kadar fazla kazazedeye bu hizmeti verirse o kadar yüksek puan (**round_score**) alacaktır.
4. İHA'ların telekomünikasyon hizmeti verebileceği azami irtifa bilgisi(**telecom_height_max**) ve asgari irtifa bilgisi(**telecom_height_min**) senaryo başında yarışma YGK'sı üzerinden yarışmacıların erişimine açılacaktır. Bu irtifadan daha yüksek irtifadaki bir İHA telekomünikasyon hizmeti veremeyecektir.
5. Bir İHA'nın telekomünikasyon hizmeti verebileceği azami kazazede sayısı aşağıdaki formülle hesaplanacaktır. Örneğin simülasyon oturumu başındaki toplam İHA sayısı 10, toplam kazazede sayısı 101 ise her bir İHA azami 11 kazazedeye telekom hizmeti verebilecektir (bölme işleminin sonucu kendinden bir büyük tam sayıya yuvarlanır).

$$THVKS_{\text{azami}} < \text{tavan} \left(\frac{\text{oturum başlangıcındaki kazazede sayısı}}{\text{oturum başlangıcındaki İHA sayısı}} \right)$$

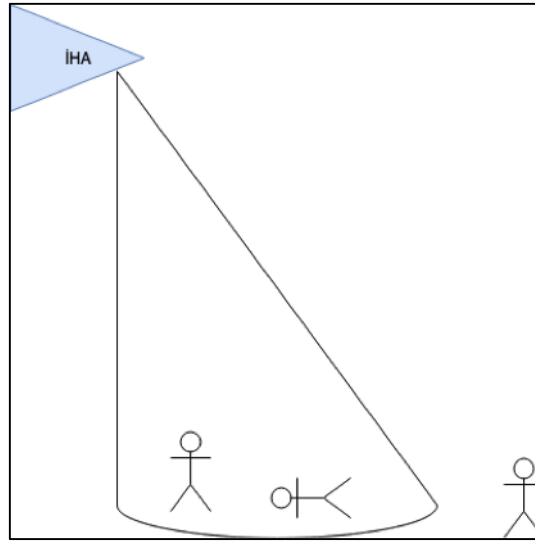
6. İHA anlık hizmet verdiği insan sayısını (**telecom_served_people_count**) yarışma YGK'sı üzerinden betiğin kontrol ettiği İHA'nın durum paketi içersinden erişebileceklerdir. İlgili bilginin elde edildiği veri paketi EK-1 İHA Durum Paketi başlığı altında incelenebilir.
7. Bir kazazede sadece 1 adet İHA'dan Telekom hizmeti alabilecek. Bundan ötürü bölgeye ilk gelen ve hizmet vermeye başlayan ilk İHA tarafından hizmet alıyor olacaktır. Eğer iki İHA'nın kapsama alanları kesişiyor ise ve kapsama alanında 1 adet İHA'nın THVKS_{azami} değerinden fazla sayıda kazazede bulunuyorsa bölgeye sonradan gelen İHA hizmet alamayan kazazedelere hizmet verebilecektir.



Şekil-3: Telekom hizmeti kapsama alanı

Yaralı Tespit ve Nakil Modülü

1. Sürü ne kadar fazla yaralıyı hastaneye taşırsa o kadar fazla puan alacaktır.
2. Bir İHA'nın üzerindeki dahili kameradan yaralı tespiti Şekil-4'deki gibidir. Kameranın görüş alanı içerisine giren kazazedenin tespit edildiği varsayılacaktır. Kameranın çalışma prensipleri hakkında bilgi almak için İHA Algılayıcı Modülü başlığı incelenmelidir.



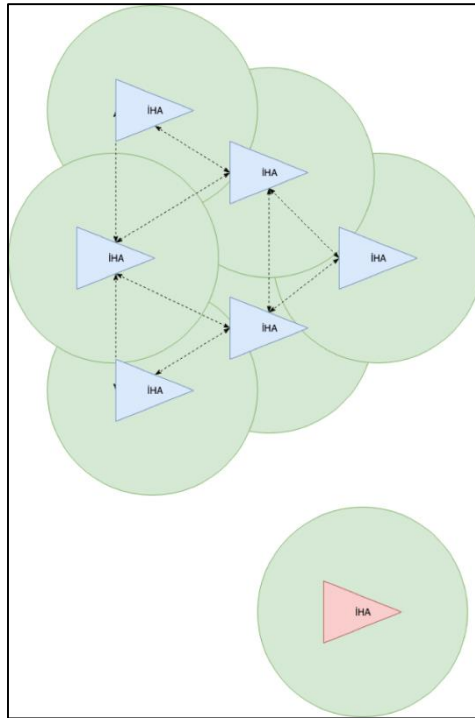
Şekil-4: Kazazede tespiti

3. İHA'nın kamera görüş alanının içerisinde olan kazazedelerin yaralı olup olmadığı bilgisi yarışma YGK'sı tarafından belirli bir port üzerinden sürekli yayın yapılacaktır. Yarışmacılar bu yayını dinleyeceklerdir. Tespit edilen insanların yaralı bilgisi, eğer yaralı ise GPS konum bilgisi yayın yapılan bilgi paketinde yer alacaktır. Yarışmacılar sadece yaralı olan kazazedeleri hastaneye taşıyabileceklerdir.
4. Yaralıların konum bilgisini alan İHA, yaralının olduğu konuma ve irtifaya azami 10 metre hata ilip gidip en az yaralı alma irtifasına (**injured_pick_up_height**) kadar alçalmalı ve bu irtifada belirli bir süre (**injured_pick_up_duration**) sabit kalmalıdır. Bu eylemi başarı ile gerçekleştiren İHA yaralıyı başarılı bir şekilde almış kabul edilir. İHA'nın alçalması beklenen en az irtifa senaryoya göre değişiklik gösterebilir. Yarışmacılar bu parametrelere senaryo başlangıcında yarışma YGK'sı üzerinden erişebileceklerdir. İlgili veri paketi örneği EK-1 Simülasyon Ortam Parametreleri başlığı altından incelenebilir.

5. İHA aldığı yaralıyı hastaneye nakledecektir. İHA aldığı yaralıyı hastaneye bırakmadan başka yaralı alamaz.
6. İHA ancak müsait olan hastaneye yaralı bırakabilir, dolu olan bir hastaneye yaralı bırakamaz. Hastanelerin kabul edebileceği yaralı bilgisi (**hospital_quota**) YGK üzerinden yayınlanacaktır. İlgili veri paketi örneği EK-1 İHA Zincir Bilgisi başlığı altından incelenebilir.
7. İHA'nın iletişim alanı içerisinde müsait bir hastane yok ise yaralıyı alan İHA'nın iletişim alanı içerisinde olan diğer İHA'ların kapsama alanına olan hastanelerin doluluk bilgisini yarışma YGK'sı İHA'ya cevap olarak dönecektir.
8. Yaralıyı taşıyan İHA hastanenin iniş pisti konumuna asgari 10 metre hata ile varıp en az yaralı bırakma irtifasına (**injured_release_height**) asgari 10 metre hata ile alçalıp en az yaralı bırakma süresi (**injured_release_duration**) kadar havada asılı kaldığında yaralıyı bırakmış sayılacaktır.

İletişim Modülü

1. Yarışmacılar sürüdeki her bir İHA'yı kendilerine oturma başında tanımlanacak TCP portu üzerinden kontrol edeceklerdir. Portların adresleri oturma başında yayınlanacaktır.
2. Her İHA'yı kontrol edecek betikler birbirinin aynısı olacaktır. Fakat betik sürüdeki hangi İHA'yı kontrol ettiğini bilecektir. İHA kendi port adresini kimlik numarası olarak kullanacaktır.
3. Her bir İHA kendi iletişim menzili içerisindeki diğer İHA'ların verilerine erişebilecektir. Bu menzil oturma bazlı değişiklik gösterecektir. Yarışmacılar bu parametreye senaryo başlangıcında YGK üzerinden erişebileceklerdir.
4. Eğer iki İHA birbirlerinin kapsama alanı içerisinde ise YGK bu İHA'lara birbirlerinin verilerine erişim hakkı verecektir (Şekil-5). İHA'lar birbirlerine komut, emir ya da bilgi paketi göndermeyeceklerdir. İHA'lar menzillerindeki diğer İHA'lar ile ilgili bilgileri YGK üzerinden alabilecektir (bkz. EK-1). Eğer bir İHA'nın kapsama alanı içerisinde başka bir İHA bulunmuyorsa, o İHA sürüdeki diğer İHA'lar hakkında hiçbir bilgiye ulaşamayacaktır. Yarışmacıların bu iletişim zincirini olabildiğince fazla İHA'yı içerecek şekilde aktif tutmaları tavsiye edilir.



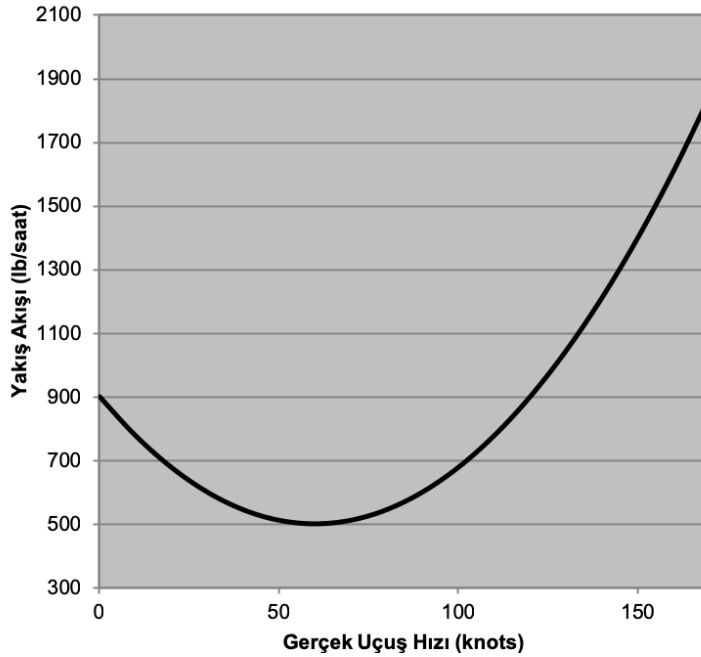
Şekil-5: İHA'lar arasındaki iletişim zinciri ve bireysel iletişim menzili

Yakıt Kontrol Modülü

1. Yarışma senaryoları belirli bir zaman kısıtlamasına tabi olmamasına rağmen, dolaylı olarak İHA'ların yakıt depo rezervlerine bağlıdır.
2. Sürüde bulunan her bir İHA'nın senaryo başlangıcında bir tam depo yakıtları olacaktır ve yakıt ikmaline izin verilmeyecektir.
3. Yarışmacıların İHA'larını yakıtları bitmeden senaryo başlangıcında kalkış yaptıkları havaalanına indirmeleri beklenmektedir. Yakıtı bittiği için havaalanına indirilemeyen İHA'lar puan kaybına sebep olacaktır.
4. İHA'lar yakıt rezervi bilgisine yarışma YGK'sı üzerinden erişebileceklerdir. Ancak her bir İHA sadece kendi rezervine ait bilgiye erişebilecektir. İHA'nın kendi iletişim menziline bulunan diğer İHA'ların yakıt rezervi bilgilerine erişmesine izin verilmeyecektir (bkz. Ek-1, örnek veri paketleri).
5. İHA'nın ağırlığının her hangi bir koşulda değişmediği varsayılacaktır. Yarışmacıların ağırlığı hesaba katmaları beklenmemektedir.
6. Jenerik bir İHA için kullanılacak olan yakıt formülü aşağıdaki verilmiştir. İHA Kontrol Modülü incelenerek, kontrol parametrelerinin yakıt kullanımına olan etkisi görülebilir. Örnek yakıt tüketimi eğrisi Şekil-6'da verilmiştir. (v: gerçek uçuş hızı (knot))

$$f(v) = \frac{(v - 60)^2}{9} + 500$$

İHA Yakıt Tüketimi



Şekil-6: Örnek yakıt tüketimi eğrisi

Puan Hesaplama Modülü

1. Yarışmacıların toplam puanı her bir senaryodan alınan puanların toplamıdır.
2. Senaryolar içerisinde icra edilmesi beklenen görevlerin her birinin belirli bir katsayı çarpanı olacaktır.
3. Her bir görevden alınan puan bu çarpanlar ile çarpılarak toplamda 100 puan üzerinden puanlandırılacaktır.
4. Telekomünikasyon Görevi katsayısı 0.3, Yaralı Tespit Etme ve Tahliye Etme Görevi katsayısı 0.35, Formasyon Uçuş Görevi katsayı 0.35 olarak belirlenmiştir.
5. Yarışmacılar senaryo sırasında anlık olarak puanlarını YGK üzerinden öğrenebileceklerdir.

Telekomünikasyon Görev Puanlaması (30 puan)

- İHA'ların üzerinde bulunan baz istasyonu sayesinde , bulunduğu konumda kaç kişiye, ne kadar süre hizmet verdiği YGK tarafından arkaplanda sürekli hesaplanır. Senaryoda verilebilecek azami hizmet süresi(Şekil-1'de bulunan yerleşim yerinde İHA'ların bulunabileceği azami süre) ve azami kişi sayısı 100 puana oranlanarak birim sürede birim kişiye verilen hizmetin puanı ortaya çıkar. Böylece yarışmacının, bu bölümden aldığı puan 100 üzerinden belirlenmiş olur.
- Puan hesaplanırken kullanılan değişkenler ve formül aşağıdaki gibidir:

Hizmet verilen insan sayısı: k

Hizmet verilen süre: t

Azami hizmet verilebilecek insan sayısı: k_{max}

Azami hizmet verilebilecek süre: t_{max}

Sürüdeki toplam İHA sayısı: h

Birim hizmet puan katsayısı: c

$$puan = \left[\sum_{i=1}^h k_i \cdot \Delta t_i \right] c$$
$$c = \frac{100}{k_{max} t_{max}}$$

- Puanlama senaryo zorluğuna göre değişmeyecektir.
- Puan hesaplama sonunda 5 puan altında alan yarışmacılar sıralamaya katılmaya hak kazanamazlar.

Yaralı Tespit Etme ve Tahliye Etme Görevi Puanlaması (35 puan)

- Senaryo sırasında İHA'ların afet bölgesinden yaralı insanları tahliye etmesi beklenmektedir.
- Bu görevdeki başarı puanının hesaplanmasında yaralıların İHA tarafından afet bölgesinden kurtarılması ve kurtarılan yaralıların uygun bir hastaneye sevk edilmesi sırasındaki başarıları hesaba katılacaktır.
- Bir kazazedeği afet bölgesinden kurtarıp ardından uygun olan bir hastaneye bırakmak yarışmacılara puan kazandıracaktır. Arada geçen süre puanlamaya dahil edilmeyecektir.
- Puan hesaplanırken kullanılan değişkenler ve formül aşağıdaki gibidir:

Başarılı bir şekilde kurtarılan kazazede sayısı: k

Kurtarılması gereken maksimum kazazede sayısı: k_{max}

$$puan = \left(\frac{k}{k_{max}} \times 100 \right)$$

- Puan hesaplama sonunda 5 puan altında alan yarışmacılar sıralamaya katılmaya hak kazanamazlar.

Formasyon Uçuş Görevi Puanlaması (35 puan)

- Senaryo başlangıcında İHA sürüsünün kılavuz hava aracını takip etmesi beklenmektedir.
- Kılavuz hava aracına ait konum bilgisi, EK-1 İHA Durum ve İHA Zincir Bilgi Paketi başlığı altında "uav_guide" başlığı altında yayınlanacaktır.

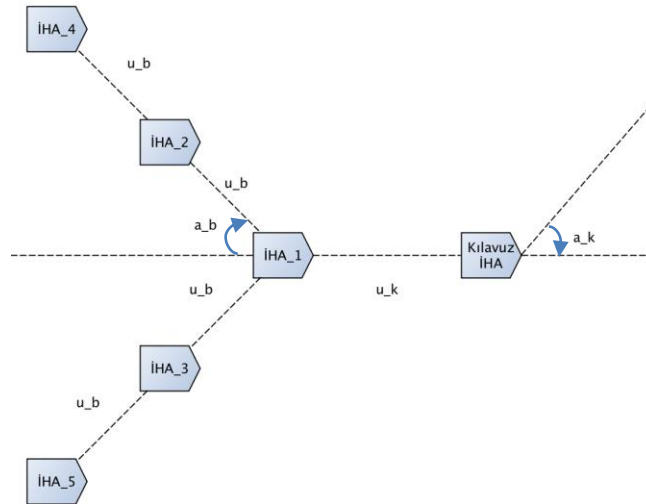
3. Bu takip sırasında sistem tarafından gönderilecek olan "dispatch" sinyali ile sürünün senaryo bazında belirlenmiş olan formasyonu oluşturarak takip görevini icra etmesi beklenmektedir. "dispatch" sinyali, EK-1 İHA Durum ve İHA Zincir Bilgi Paketi başlığı altında "uav_guide" anahtar kelimesi içerisinde bulunmaktadır.
4. Bu aşamada, sürünün başarılı bir şekilde formasyon değiştirmesi 10 puandır. Uçuş süresi boyunca formasyon koruma 10 puan (bkz. Madde 8) ve GPS sinyali bozulduktan sonra en son uçulmakta olan formasyonu tekrar almak 15 puan (bkz. Madde 8) olacaktır. GPS sinyalinin ne zaman bozulacağı İHA'lara bildirilmeyecektir ancak bozulma bütün İHA'lar için aynı anda gerçekleşecektir.
5. Sürünün "ok başı" ve "dikdörtgen prizma" olmak üzere iki farklı formasyonda uçuşması istenecektir. YGK an itibariyle geçerli formasyonu bilgi paketinde (EK-1) sürekli yayınlayacaktır. Bilgi paketindeki geçerli formasyon tipi bilgisini takip edip değişikliklere uyum sağlamak yarışmacının sorumluluğundadır.
6. Ok başı formasyonunun dizilimi aşağıdaki gibidir. Bu formasyonda sürünün tüm bireyleri aynı irtifadadır. Bu formasyonda kılavuza en yakın İHA, kılavuzu (**uav_guide_follow_distance**) mesafesinden takip edecektir.

Komşu bireyler arası uzaklık: u_b

"ok başı" formasyon tepe açısı yarı: a_b

Kılavuz takip mesafesi: u_k

Kılavuz baş açısı (pusulaya göre): a_k



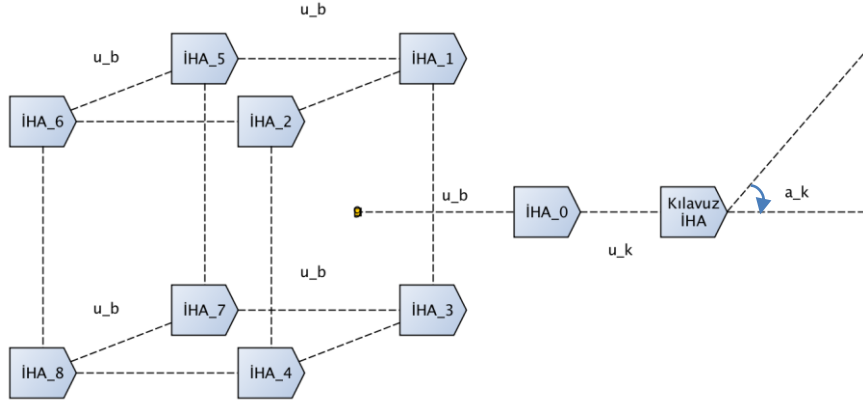
Şekil-7: Ok başı formasyon uçuşu

7. "Dikdörtgen prizma" formasyonu Şekil-8'de gösterilmiştir. Dikdörtgen prizma "2 x 2 x n" boyutunda 3 boyutlu bir matris yapısından oluşmaktadır.

Komşu bireyler arası uzaklık: u_b

Kılavuz takip mesafesi: u_k

Kılavuz baş açısı (pusulaya göre): a_k



Şekil-8: Dikdörtgen prizma formasyon uçuşu

8. İHA'nın formasyon uçuşunda bulunması gereken noktaya olan uzaklığı, geçen zaman ve ceza katsayısı ile hata oranı hesaplanır. Çıkan oran doğrultusunda alabileceği maksimum puandan ceza miktarı düşülür. Sürünün formasyonu ne kadar başarılı oluşturduğunu ve koruduğunu hesaplamak için aşağıdaki formül kullanılacaktır:

Formasyon halinde uçulan süre: t

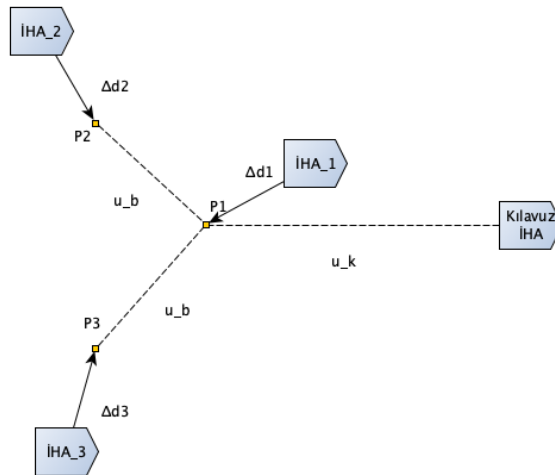
İHA'nın xyz düzleminde bulunması gereken nokta ile anlık bulunduğu nokta arasındaki uzaklık: d

Formasyon ceza puanı ölçeklendirme katsayısı: c

$$hata = \sum_{i=0} \Delta d_{xyz_i} \Delta t_i$$

Formasyon aşamasından alınabilecek azami puan: $puan_{azami}$

$$puan = puan_{azami} - hata \times c$$



Şekil-9: Formasyon uçuşu hata hesaplanması. P_n İHA'nın olması gereken noktayı, Δd_n ise konumundaki hata miktarını göstermektedir.

Ceza Puanlaması

1. Yarışmacıların sürülerinde bulunan herhangi bir İHA uçuşa yasak bölgeye giriş yaparsa aşağıda belirtildiği şekilde ceza puanı yansıtılır:
 - a. İlk ve tek ihlal için -5 puan
 - b. Birden fazla ihlal yapıldıysa bir kerelik -20 puan
2. Yarışmacıların sürülerinde bulunan İHA'ları güvenli bir şekilde iniş bölgesine indirmeleri beklenmektedir. Eğer herhangi bir İHA yakıtı bitip yere düşerse, sürüdeki diğer İHA'lar ile çarpışırsa veya sahnede bulunan yapılara çarpıp düşerse hanesine eksi puan olarak aşağıdaki şekilde yansıtılır. Çarpışan İHA'lar düşmüş olarak kabul edilir.
 - a. Bir İHA düşerse -10 puan.
 - b. Birden fazla İHA düşerse bir kerelik -25 puan.

Senaryo Kontrol Modülü

1. Yarışmacıların bu modüle erişimi olmayacaktır, bu modül hakkındaki bilgiler bilgilendirme amaçlı verilmiştir.
2. Yarışmanın gerçekleşeceği senaryoları kontrol eden modüldür.
3. Senaryo içerisinde icra edilmesi beklenen görevleri yönetir.
4. Senaryo başlangıcında İHA sürüsüne kılavuzluk eden hava aracını kontrol eder.
5. Yarışmacılar senaryo ile ilgili bütün parametrelere yarışma YGK'si üzerinden erişebileceklerdir. Yarışmacılar EK-1 başlığı altında bulunan Simülasyon Ortamı Parametre Paketi'ni inceleyerek örnek veri paketini görebilirler.
6. Senaryo dünyasında bulunan hastanelerin, kaçınılması gereken yüksek binaların ve uçuşa yasaklı bölgelerin(denied_zones) konumuna ve yüksekliğine EK-1 başlığı altında bulunan Simülasyon Ortamı Parametre Paketi üzerinden erişebileceklerdir.
7. Uçuşa yasaklı bölgeler konveks veya konkav şekiller olabilir. İlgili veri paketinde (bkz. EK-1 Simülasyon Ortamı Parametre Paketi) şeklin köşe koordinatları sıralı olarak verilecektir. Yarışmacılar uçuşa yasaklı bölgeleri, koordinatlar arasında doğru parçaları çizerek anlamlandırıp, bu bölgelerden kaçınmalıdır. (Python'da Shapely kütüphanesi kullanılarak bu anlamlandırma yapılabilmektedir)
8. Uçuşa yasaklı bölgelerin(denied_zones) belirli bir irtifa ile sınırlandırılmamıştır. İHA'lar bu bölge üzerinde uçmamalıdır.

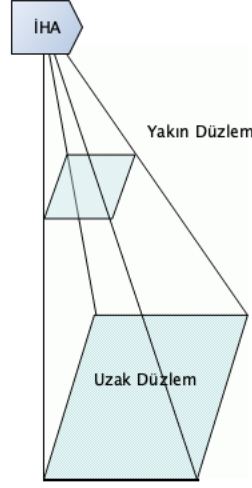
İHA Uçuş Kontrol Modülü

1. Yarışmacıların İHA kontrol yazılımlarının, simülasyon içerisinde bulunan İHA'ları dinamik olarak kontrol etmesini sağlayan modüldür.
2. Yarışmacılar bu modül aracılığı ile İHA'lara uçuş komutları göndererek İHA'ları kontrol edeceklerdir. Bu komutlar, her bir İHA için belirlenmiş olan TCP/UDP portları üzerinden gönderilecektir.
3. Yarışmacıların İHA'yı kontrol edebilmesi için, İHA'ya atanmış olan TCP/UDP portu üzerinden komut göndermeleri gereklidir. Yarışma YGK'si bu komutları dinleyerek sahnede bulunan İHA modelinin aerodinamik hesaplamalarını yapar ve modeli sahnede hareket ettirir.
4. Yarışmacıdan beklenen referans komutlar aşağıdaki gibidir:
 - Gövde eksenine göre, X eksenindeki uçuş hızını belirlemek için "x.speed" anahtar kelimesi ile komut gönderilir.
 - Gövde eksenine göre, Y eksenindeki uçuş hızını belirlemek için "y.speed" anahtar kelimesi ile komut gönderilir.

- Tırmanması beklenen irtifayı belirlemek için “altitude” anahtar kelimesi ile komut gönderilir.
 - İHA'nın baş açısını belirlemek için “heading” anahtar kelimesi ile komut gönderilir.
5. X eksenindeki uçuş hızının cinsi “knot” olarak belirlenmiştir. “x.speed” parametresi İHA'nın gövde eksenine göre x eksenindeki hızını belirleyecektir. Yarışma YGK'sı ise bu değerleri aerodinamik modülden geçirerek İHA'nın hareket etmesini sağlayacaktır.
 6. Y eksenindeki uçuş hızının cinsi “knot” olarak belirlenmiştir. “y.speed” parametresi İHA'nın gövde eksenine göre Y eksenindeki hızını belirleyecektir. Yarışma YGK'sı ise bu değerleri aerodinamik modülden geçirerek İHA'nın hareket etmesini sağlayacaktır.
 7. Yarışmacılar İHA'nın tırmanacağı/alçalacağı irtifayı yarışma YGK'sına göndererek İHA'nın irtifasını belirleyeceklerdir. İrtifanın cinsi “feet” olarak belirlenmiştir. Yarışmacıların İHA'yı istedikleri irtifada sabitlemeleri için ekstra bir işlem yapmaları gerekmemektedir. Ancak çevresel faktörlerden ötürü İHA'nın irtifasını koruması sırasında sapmalar meydana gelebilir. Olası her türlü çarpışmayı engellemek(collision avoidance) yarışmacıların sorumluluğundadır.
 8. İHA'ların aerodinamik yapısından ötürü, sistem üzerinde dinamik etkileşim (coupling) vardır. Örneğin, “x.speed” referans değeri güncellendiğinde İHA'nın mevcut “heading” açısında bir miktar sapma, mevcut irtifasında (altitude) bir miktar değişim oluşabilecektir. Bu etkileşimi göz önünde bulundurmak yarışmacının sorumluluğundadır.
 9. İHA uçuş kontrol modülü x ve y eksenlerinde hız kontrolcüsü ve z-ekseninde pozisyon kontrolcüsünden oluşmaktadır.
 10. İHA'nın gerçek uçuş hızı yakıt tüketimi hesaplanırken doğrudan kullanılmaktadır. Yakıt tüketimi ile ilgili daha detaylı bilgi için İHA Yakıt Kontrol Modülü başlığı incelenmelidir.
 11. Baş açısı parametresinin birimi ise “derece” olarak belirlenmiştir. Bu değer gerçek kuzey baz alınarak ortaya çıkan pusula derecesidir. Örneğin, mevcut “heading” değeri 30 derece olan bir İHA'nın “heading” referans değeri 120 derece olarak belirlenirse, İHA saat yönünde 90 derece dönecektir.

İHA Algılayıcı Modülü

1. İHA'ların senaryoda belirtilen görevleri icra edebilmesi için gerekli olan algılayıcıları kontrol eden modüldür.
2. Bu algılayıcılar içerisinde GPS ve sanal kamera sensörleri bulunmaktadır. Yarışmacılar İHA'ların üzerinde bulunan bu algılayıcılara ait bilgilere bağlantı sağlandığı taktirde erişebileceklerdir. İletişim kanalı olarak TCP/UDP kanalları kullanılacaktır.
3. Her bir İHA'nın üzerinde bir adet sanal kamera modülü bulunacaktır. Bu sanal kamera modülü yere doğru bakacak olup, yarışmacıların İHA'larına çevre farkındalığı kazandırmak için kullanması beklenecektir.
4. Normalde yarışmacılardan kamera görüntüsü üzerinde görüntü işleme yapmaları beklenir. Ancak yarışmanın amacı sürü algoritması geliştirmek olduğu için yarışmacılara ham kamera görüntüsü gönderilmeyecektir.
5. Yarışma YGK'si tarafından sanal kameranın görüş açısına giren unsurların bilgileri gönderilecektir.
6. Sanal kameranın görüş alanı hesaplanırken, önce İHA'nın sanal azami kamera çalışma yüksekliğinde (**logical_camera_height_max**) ve asgari kamera çalışma yüksekliğinde (**logical_camera_height_min**) olup olmadığı hesaplanır. Eğer İHA, kamera çalışma yüksekliğinin üzerinde değilse sanal kameranın görüş alanındaki (Şekil-10) kazazede konum ve sağlık bilgileri (bkz. EK-1 İHA Durum Bilgisi) YGK üzerinden yayınlanacaktır.
7. Kullanılan sanal kameranın yatay görüş açısı (**logical_camera_horizontal_fov**) ve kamera yunuslama açısı (**logical_camera_pitch**) yarışma parametrelerinden erişilebilecektir.



Şekil-10: Sanal kamera sensörü

8. Simülasyon ortamı içerisinde, yarışmacılar İHA'ların ve diğer unsurların konumlarını GPS bilgisi olarak alacaklardır.
9. Simülasyon içerisinde üretilen GPS verileri üzerinde belirli bir miktar gürültü bulunmaktadır. Ayrıca sanal kamera görüntüsünden elde edilen konum bilgileri, kamera görüntüsü üzerinde oluşabilecek potansiyel gürültüden ötürü bir miktar hata ile İHA'ya yayınlanacaktır.
10. Her bir İHA'nın üzerinde bulunacak olan GPS modülü sayesinde yarışmacıların geliştirmiş olduğu yazılımlar, kontrol ettikleri İHA'nın ve iletişim menzili içerisinde bulunan sürüdeki diğer İHA'ların konumlarını (enlem, boylam, irtifa) öğrenebileceklerdir. Gerekli olması halinde eksen dönüşümlerini yapmak yarışmacıların sorumluluğundadır.

8 YARIŞMA KURALLARI

Yarışma YGK Kısıtlamaları

1. Yarışmacıların İHA'ları kontrol etmek ve İHA'lar hakkında bilgi almak dışında yarışma simülasyonunun gerçekleştiği sisteme müdahale etmeleri yasaktır.
2. Her bir İHA yalnızca bir adet betik ile kontrol edilmelidir, İHA'ları kontrol eden betikler arasında iletişim yasaktır. Bütün betikler ise bir adet Docker konteyneri içerisinde çalıştırılacaktır. Eğer tek betik ile birden fazla İHA'nın kontrol edildiği saptanırsa yarışmacı yarışmadan diskalifiye edilecektir.
3. Yarışmacılar İHA'ya komut gönderirken ve İHA'dan bilgi alırken sadece yarışma YGK'sını kullanmalıdır. Farklı yöntemler ile ROS veya Gazebo ortamına müdahale tespit edilirse yarışmacı yarışmadan men edilecektir.
4. İHA'lar arası iletişim daha önceki başlıklarda detaylı açıklanmıştır. Sürü algoritması geliştirilmesindeki amaç, sürünün merkezi olmayan bir şekilde yönetilmesidir. Bu sebepten ötürü, İHA'lar arasında yarışma YGK'sı dışında izin verilmeyen bir yöntem ile oluşturulacak herhangi bir iletişim sistemine izin verilmeyecektir. Böyle bir iletişim kurulduğu tespit edildiği takdirde yarışmacı yarışmadan men edilecektir.
5. Örnek Docker konteyneri ve örnek kodlar yarışmacılar ile yarışma takvimi uyarınca paylaşılacaktır.

Yarışma Ortamı Kısıtlamaları

1. Bütün yarışma ortamı önceden hazırlanmış olan bir Docker konteyneri olarak belirlenmiştir. Bu konteynerin iletişime izin verdiği TCP portları da önceden belirlenmiştir. Yarışma sırasında da bu portlar kullanılacağı için, yarışmacıların yazılım geliştirme aşamasında port sayısını artırma veya port adreslerini değiştirme gibi işlemler yapmamaları tavsiye edilir.
2. Yarışmacıların yazılımlarını Docker konteyneri olarak teslim etmesi beklenmektedir. Böylece geliştirme yapacakları kütüphaneleri ve iskelet yapılarını kendilerinin belirleme hakları bulunmaktadır. Ancak yarışma nezdinde haksız avantaj sağlayacak herhangi bir yardımcı açık kaynaklı yazılım kullanıldığı tespit edilirse, yarışmacılar diskalifiye edilecektir. Bu sorunu ortadan kaldırmak için yarışmacılar kullanacakları açık kaynak kodları/kütüphaneleri/altyapıları yarışmanın iletişim kanalından yarışma komitesine bildirmeleri ve netleştirmeleri beklenmektedir.

9 EK-1

İHA Durum ve İHA Zincir Bilgi Paketi

```
{
  "active_uav":{
    "location": "39.924980, 32.836794",
    "altitude": "1000",
    "heading": "30.0",
    "speed":{
      "x": "60.0",
      "y": "10.0"
    }
  },
  "fuel_reserve": "%100.0",
  "sensors":{
    "logical_camera":{
      "detected":[
        {
          "type": "casualty",
          "status": "injured",
          "pose": "39.924980, 32.841231"
        },
        {
          "type": "casualty",
          "status": "healthy",
          "pose": "39.944980, 32.836794"
        }
      ]
    }
  },
  "equipments":{
    "telecom_beacon":{
      "telecom_served_people_count": 5
    }
  },
  "uav_link":{
    "uav_1":{
      "location": "39.924980, 32.836794",
      "altitude": "1000",
      "heading": "30.0",
      "speed":{
        "x": "60.0",
        "y": "10.0"
      }
    },
    "uav_2":{
      "location": "39.924980, 32.836794",
      "altitude": "1000",
      "heading": "30.0",
      "speed":{
        "x": "60.0",
        "y": "10.0"
      }
    },
    "uav_3":{
      "location": "39.924980, 32.836794",
      "altitude": "1000",
      "heading": "30.0",
      "speed":{
        "x": "60.0",
        "y": "10.0"
      }
    }
  },
  "hospitals_in_range":[
    {
      "location": "39.933155, 32.822292",
      "hospital_quota": 10
    }
  ],
  "uav_guide":{
    "location": "39.924980, 32.836794",
    "altitude": "1000",
    "heading": "30.0",
    "speed":{
      "x": "60.0",
      "y": "10.0"
    }
  },
  "dispatch": False,
  "uav_formation": {
    "type": "arrow",
    "u_b": "25.0",
    "u_a": "30.0",
    "u_k": "25.0",
    "a_k": "45.0"
  }
}
```

Simülasyon Ortamı Parametre Paketi

```
{
  "sim_env_parameters":{
    "level_difficulty":1,
    "injured_pick_up_duration":30,
    "injured_pick_up_height":"15.0",
    "injured_release_duration":30,
    "injured_release_height":"15.0",
    "uav_communication_range":"1000.0",
    "uav_count":10,
    "telecom_radius":"500.0",
    "telecom_height_max":"750.0",
    "telecom_height_min":"50.0",
    "world_width":"5000.0",
    "world_length":"5000.0",
    "logical_camera_height_max":"500.0",
    "logical_camera_height_min":"50.0",
    "logical_camera_pitch":"60.0",
    "logical_camera_horizontal_fov":"60.0"
  },
  "special_assets":[
    {
      "type":"hospital",
      "location":"39.933155, 32.822292",
      "height":40
    },
    {
      "type":"hospital",
      "location":"39.920826, 32.842875",
      "height":40
    },
    {
      "type":"tall_building",
      "location":"39.920826, 32.842875",
      "height":70
    }
  ],
  "denied_zones":[
    {
      "zone_1":[
        "39.915393 32.845461",
        "39.918012 32.846343",
        "39.916765 32.850036",
        "39.915037 32.848852"
      ]
    }
  ]
}
```

İHA Kontrol Paketi

```
{
  "uav_control":{
    "x_speed":"60.0",
    "y_speed":"10.0",
    "altitude":"1000.0",
    "heading":"90.0"
  }
}
```