

## II. İlgili Çalışmalar

### 1. Otomatik Kontrol Sistemleri:

- [9], radyo frekansı darbeleri ve Frenet-Serret denklemlerini kullanarak İHA sürüleri için otomatik bir kontrol sistemi önerir. Ancak, önceden planlanmış görevlerle ölçeklenebilir sürülere odaklanmaz.

### 2. Koordinasyon Protokolleri:

- [10], coğrafi yönlendirme kullanarak arama görevleri için bir koordinasyon protokolü sunar ve 2D simülasyonlarla değerlendirilir.
- [11], sürü koordinasyonu için 3G/4G altyapısını ve Ölçeklenebilir Veri Teslim Katmanı (SDDL) kullanır.
- [12], afet bölgelerinde kablosuz iletişim kurmak için İHA'lar üzerinde özerk ajanlar kullanır ve merkezi olmayan işbirlikçi kontrolü hedefler.

### 3. İletişim Topolojileri:

- [13], İHA sürüleri için bağlantı ve kapsam alanını maksimize etmek amacıyla ad-hoc ağlar önerir.
- [14], Uçan Ad-Hoc Ağlar (FANET'ler) ile ilgili zorlukları kapsamlı bir şekilde inceler ve ölçeklenebilirlik, gecikme ve uyum sağlama gibi konuları ele alır.

### 4. Mikro İHA Sürüleri ve Formasyon Kontrolü:

- [15], mikro İHA sürüleri için Zigbee adaptörleri kullanır ancak laboratuvar ortamlarıyla sınırlıdır.
- [16], 3D İHA sürüleri için sanal bir yapı tabanlı formasyon kontrolörü tanımlar ve simülasyonlarla doğrulanır.

### 5. Hücresel Ağlar ve Stabilitate:

- [17], hücresel ağ gecikmelerinin sürü stabilitesi üzerindeki etkisini analiz eder ve stabilite için maksimum gecikme sınırlarını belirler.

### 6. Sanal Lider Yapıları ve Zamanla Değişen Formasyonlar:

- [18], merkezi olmayan İHA sürüleri için sanal bir lider tabanlı kontrolör kullanarak katı formasyonlar sağlar.
  - [19], zamanla değişen formasyonlar için bir anahtar etkileşim topolojisi benimser ve gerçek dört pervaneli İHA'larla doğrulanır.
-

### III. ArduSim Simülatörüne Genel Bakış

**MUSCOP Protokolü**, Fabra ve arkadaşları tarafından geliştirilen ve çevrimiçi olarak Apache License 2.0 altında sunulan **ArduSim** adlı çoklu İHA (İnsansız Hava Aracı) simülatörü/emülatörü kullanılarak geliştirilmiş ve değerlendirilmiştir.

ArduSim, 256 adet çoklu pervaneli İHA'nın fiziksel davranışlarını yüksek doğrulukla taklit edebilir ve sanal Wi-Fi bağlantıları üzerinden İHA'lar arasındaki iletişimi simüle eder.

#### ArduSim'in Temel Özellikleri

- Gerçek İHA'lara Kolay Protokol Entegrasyonu:**
    - Gerçek açık kaynaklı İHA'lar, MAVLink iletişim protokolü kullanılarak kontrol edilir.
    - ArduSim, sanal İHA'ları MAVLink ile yönetir. Bu protokol, yayın-abone ve noktadan noktaya iletişim modellerini birleştirir.
    - Gerçek İHA'lara, bir Raspberry Pi ve kablosuz adaptör eklenerek ArduSim protokolleri kolayca entegre edilebilir.
  - Yumuşak Gerçek Zamanlı Simülasyon:**
    - Tüm İHA'lar neredeyse gerçek zamanlı olarak çalışır, bu da protokol geliştirme ve hata ayıklama sürecini hızlandırır.
  - Ölçeklenebilirlik:**
    - Yüksek performanslı bir bilgisayarda (Intel Core i7-7700, 32 GB RAM) 100 İHA gerçek zamanlı, 256 İHA ise yumuşak gerçek zamanlı olarak simüle edilebilir.
  - İHA'dan İHA'ya İletişim:**
    - Simülasyon sırasında İHA'lar, 802.11a standardına dayalı sanal bağlantılar üzerinden iletişim kurar.
    - Protokol gerçek cihazlara dağıtıldığında, UDP yayınları kullanılarak ad-hoc ağ üzerinden iletişim sağlanır.
  - Kapsamlı API:**
    - ArduSim, kalkış, görev kontrolü, iniş gibi İHA davranışlarını yönetmek için eksiksiz bir API sunar.
  - Dağıtım Desteği:**
    - ArduSim, simülatör olarak veya gerçek İHA'larda çalıştırılabilir. Ayrıca, bir dizüstü bilgisayarda "PC Companion" olarak çalıştırılarak protokolün başlatılması ve denetlenmesi kolaylaştırılır.
  - Otomatik Çarpışma Tespiti:**
    - Simülasyon sırasında İHA'lar arasında çarpışma olursa kullanıcıya bildirilir, bu da protokol tasarımının iyileştirilmesine yardımcı olur.
  - Kapsamlı Veri Kaydı:**
    - Simülasyon veya gerçek uçuş sonunda, İHA'ların izlediği yol (koordinatlar, yön, hız, ivme vb.) kaydedilir. Bu veriler, OMNeT, NS2 ve Google Earth formatlarında da saklanır.
-

## IV. MUSCOP Protokolü

**MUSCOP Protokolü**, önceden planlanmış bir görevi takip ederken İHA (İnsansız Hava Aracı) sürülerinin koordinasyonunu sağlamak için geliştirilmiştir. Bu protokol, ArduSim simülatörü kullanılarak geliştirilmiş ve gerçek İHA'lara doğrudan uygulanabilir özelliktedir. Protokol, master-slave (lider-izleyici) modeline dayanır ve sürünün uçuş formasyonunu kararlı bir şekilde korumasını hedefler.

### A. Protokol Genel Bakışı

- **Master-Slave Modeli:** Sürüdeki lider İHA (master), görev sırasında diğer İHA'ları (slave) senkronize eder. Her yol noktasında (waypoint), master İHA, slave İHA'ların bir sonraki harekete geçmesini bekler.
- **Görev Planlaması:** Kalkış öncesinde, master İHA, slave İHA'lara göreceli konumlarını dikkate alarak ayarlanmış bir görev planı gönderir. Tüm İHA'lar aynı hızda hareket eder, böylece uçuş formasyonu görev boyunca korunur.
- **İletişim:** Her İHA, **Talker Thread** (konuşmacı iş parçacığı) ve **Listener Thread** (dinleyici iş parçacığı) olmak üzere iki iş parçacığı kullanır. Master İHA, komutlar gönderirken, slave İHA'lar bu komutları alır ve onay mesajları gönderir.

### B. Sonlu Durum Makinesi (Finite State Machine)

Protokolün davranışı, master ve slave İHA'lar için ayrı sonlu durum makineleri (FSM) ile yönetilir. Bu makineler, İHA'ların farklı durumlar arasında geçiş yapmasını ve birbirleriyle senkronize olmasını sağlar.

#### Master FSM:

1. **Hazırlık Durumu:** Master İHA, kalkış öncesinde tüm slave İHA'ların hazır olduğunu doğrular.
2. **Görev Başlatma:** Görev başlatıldığında, master İHA, her bir waypoint için komutları slave İHA'lara iletir.
3. **Bekleme Durumu:** Master İHA, tüm slave İHA'ların belirlenen waypoint'e ulaştığını doğrular.
4. **Görev Tamamlama:** Tüm waypoint'ler tamamlandığında, iniş komutunu iletir ve görev sonlandırılır.

#### Slave FSM:

1. **Hazırlık Durumu:** Slave İHA, master İHA'dan gelen başlangıç komutlarını bekler.
  2. **Hareket Durumu:** Gelen komut doğrultusunda waypoint'e hareket eder.
  3. **Onay Gönderimi:** Waypoint'e ulaştığında, master İHA'ya onay mesajı gönderir.
  4. **Görev Tamamlama:** Master İHA'nın iniş komutunu aldıktan sonra görevi sonlandırır.
-

## V. Deneysel Sonular

MUSCOP protokolü, ArduSim simülatörü kullanılarak kapsamlı bir şekilde test edilmiştir. Testler sırasında aşağıdaki performans ölçütleri değerlendirilmiştir:

### A. Görev Tamamlama Süresi

- **Sonuç:** MUSCOP protokolü, 50 İHA ile yapılan testlerde görev tamamlama süresini optimize ederek senkronize hareket sağladı.

### B. Çarpışma Önleme

- **Sonuç:** Tüm testlerde, MUSCOP protokolü sayesinde İHA'lar arasında çarpışma yaşanmadı.

### C. Ölçeklenebilirlik

- **Sonuç:** MUSCOP, 256 İHA'ya kadar ölçeklenebilir bir yapı sundu. Performans düşüşü minimum seviyede kaldı.

### D. İletişim Gecikmeleri

- **Sonuç:** Wi-Fi tabanlı iletişimde, gecikmelerin protokolün stabilitesine etkisi gözlemlendi ve maksimum gecikme sınırları belirlendi.

---

## VI. Sonuç ve Gelecek Çalışmalar

MUSCOP protokolü, önceden planlanmış görevleri senkronize bir şekilde yerine getirebilen ölçeklenebilir ve güvenilir bir İHA sürü protokolü olarak geliştirilmiştir.

### Gelecek Çalışmalar:

1. **Dinamik Görev Atama:** Görev sırasında değişen koşullara göre İHA'ların rollerinin dinamik olarak yeniden atanması.
2. **Enerji Verimliliği:** İHA'ların enerji tüketimlerini optimize eden algoritmaların entegrasyonu.
3. **3D Formasyonlar:** Zamanla değişen üç boyutlu formasyonlar için protokolün genişletilmesi.
4. **Gerçek Uygulamalar:** Protokolün gerçek dünyadaki afet yönetimi ve lojistik uygulamalarında test edilmesi.

## Kaynakça

- <https://ieeexplore.ieee.org/document/9066837>