

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ÇOK ETMENLİ SİSTEMLERİN İŞ BİRLİĞİ KONTROLÜNDE KALMAN
FİLTRESİ UYGULAMALARI**

YÜKSEK LİSANS TEZİ

Orhun Esat ŞAFAK

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

TEMMUZ 2018

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ÇOK ETMENLİ SİSTEMLERİN İŞ BİRLİĞİ KONTROLÜNDE KALMAN
FİLTRESİ UYGULAMALARI**

YÜKSEK LİSANS TEZİ

**Orhun Esat ŞAFAK
(518131021)**

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Gülay ÖKE GÜNEL

TEMMUZ 2018

İTÜ, Fen Bilimleri Enstitüsü'nün 518131021 numaralı Yüksek Lisans Öğrencisi Orhun Esat ŞAFAK, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “ÇOK ETMENLİ SİSTEMLERİN İŞ BİRLİĞİ KONTROLÜNDE KALMAN FİLTRESİ UYGULAMALARI” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Gülay ÖKE GÜNEL**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Mehmet Turan SÖYLEMEZ**
İstanbul Teknik Üniversitesi

Dr. Öğr. Üyesi Figen ÖZEN
Haliç Üniversitesi

Teslim Tarihi : 2 Mayıs 2018
Savunma Tarihi : 16 Temmuz 2018





Nişanlıma ve aileme,



ÖNSÖZ

Tezimi gerçekleştirmemde emek harcayıp vaktini ayıran danışmanım Doç. Dr. Gülay ÖKE GÜNEL'e ve motive olmamı sağlayan bütün arkadaşlarıma teşekkürlerimi sunuyorum.

Her zaman benim yanımda olan değerli nişanlıma ve aileme sonsuz şükranlarımı sunuyorum.

Mayıs 2018

Orhun Esat ŞAFAK
(Makine Mühendisi)



İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
ÇİZELGE LİSTESİ.....	xi
ŞEKİL LİSTESİ.....	xiii
ÖZET.....	xv
SUMMARY	xvii
1. GİRİŞ.....	1
1.1 Amaç ve Kapsam	2
1.2 Çalışmanın Ana Hatları	2
2. ÇOK ETMENLİ SİSTEMLERDE İŞ BİRLİĞİ	5
2.1 Çok Etmenli Sistem Yapısında İş Birliğinin Anlamı	5
2.2 Graf Teorisi	7
2.3 İş Birliği Algoritmasının Temelleri.....	10
2.4 Çok Etmenli Sistem Uygulamaları.....	19
3. KALMAN FİLTRESİ.....	25
3.1 Olasılığa Giriş ve Normal Dağılımlar	25
3.2 Lineer Sistemlerde Tahmin Yöntemlerinin Temelleri	34
3.3 Kalman Filtresi Algoritması.....	36
3.4 Genişletilmiş Kalman Filtresi.....	41
4. ÇOK ETMENLİ SİSTEMLERDE KALMAN FİLTRESİ	
UYGULAMALARI.....	53
4.1 Çift Entegrator Sistemi.....	54
4.1.1 Sisteme ait parametre ve denklemler	54
4.1.2 Sisteme ait simülasyon sonuçları ve yorumlanması.....	58
4.2 Harmonik Osilator Sistemi.....	61
4.2.1 Sisteme ait parametre ve denklemler	62
4.2.2 Sisteme ait simülasyon sonuçları.....	65
4.3 Sinüzoidal Sistem.....	70
4.3.1 Sisteme ait parametre ve denklemler	70
4.3.2 Sisteme ait simülasyon sonuçları ve yorumlanması.....	73
5. SONUÇLAR	77
KAYNAKLAR	81
ÖZGEÇMİŞ.....	85



ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : Çift entegratör sistemi için L_2 norm	79
Çizelge 2.1 : Harmonik osilator sistemi için L_2 norm	79
Çizelge 2.2 : Sinozoidal sistem için L_2 norm.	80





ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Bir graf topolojisi[10].....	7
Şekil 2.2 : Yönsüz (undirected) graf.....	8
Şekil 2.3 : Kontrol yöntemleri	11
Şekil 2.4 : Üç farklı graf için iletişim topolojiler[8].....	13
Şekil 2.5 : Çok sayıda elemandan oluşan ağ yapısı.[16]	15
Şekil 2.6 : Her bir elemanın ihtiyaç duyduğu bilgi alışverişini sağlayan iletişim ağı gösterilmektedir[23].	21
Şekil 2.7 : Ağ yapısındaki her bir eleman kendi ekseninde dönerek doğru açığı bulmaya çalışmaktadır.....	22
Şekil 2.8 : Ağ yapısındaki elemanlar, canlı varlıktan gelen sinyal ile komşularından aldığı pozisyon bilgilerini bir araya getirerek belirli bir sürüş açısı elde etmeye çalışmaktadır[23]..	22
Şekil 2.9 : Her bir eleman komşularından aldığı bilgiyi de kullanarak belirli bir koordinat doğrultusunda canlı varlığa doğru gitmektedir[23]	23
Şekil 2.10 : Ağ yapısındaki elemanlar verilen görevi takım halinde çalışarak iş birliği içerisinde yerine getirmektedir[23]	23
Şekil 3.1 : Hatta beklenen süre için PDF (probability density function)[25]......	28
Şekil 3.2 : Binom dağılımdaki olasılıkların gösterimi[25].	30
Şekil 3.3 : Poisson dağılımda olasılıkların gösterimi[25]......	30
Şekil 3.4 : Üstel yoğunlukların gösterimi.	31
Şekil 3.5 : Belirli parametrelere göre normal yoğunluğun gösterimi.	32
Şekil 3.6 : Normal eğri[26]......	33
Şekil 3.7 : X değişkeninin iki değer aralığı arasındaki puan veya gözlem oranlarının gösterimi[26].	33
Şekil 3.8 : Standart normal eğri altında kalan alanın yüzdesel olarak gösterimi[26].	34
Şekil 3.9 : Bir gözlemci için blok diyagramın gösterimi[27]......	35
Şekil 3.10 : Kalman filtresi döngüsü.	40
Şekil 3.11 : Kalman filtresinin bütünü gösteren diyagram[24].	41
Şekil 3.12 : Genişletilmiş Kalman filtresi denklemleri[24]......	44
Şekil 3.13 : Lineer olmayan akım/gerilim ilişkisi.	47
Şekil 4.1 : Sistemdeki elemanların bilgilerinin oluşturulması.....	55
Şekil 4.2 : Sistemdeki elemanların komşuluk ilişkisinin oluşturulması.....	55
Şekil 4.3 : Sistemde bulunan elemanlar için kontrol girdilerinin oluşturulması	56
Şekil 4.4 : Sisteme gürültülerin eklenmesi.	57
Şekil 4.5 : Kalman filtresi uygulanması	57
Şekil 4.6 : Çift entegratör sistemi için gürültüsüz ortam (eleman sayısı :30).....	58
Şekil 4.7 : Çift entegratör sistemi için gürültüsüz ortam (eleman sayısı :20).....	59
Şekil 4.8 : Çift entegratör sistemi için gürültüsüz ortam (eleman sayısı :10).....	59
Şekil 4.9 : Çift entegratör sistemi için gürültülü ortam (eleman sayısı :30).....	59

Şekil 4.10 : Çift entegratör sistemi için gürültülü ortam (eleman sayısı :20).....	60
Şekil 4.11 : Çift entegratör sistemi için gürültülü ortam (eleman sayısı :10).....	60
Şekil 4.12 : Çift entegratör sistemi için Kalman filtresi uygulaması (eleman sayısı :30).....	60
Şekil 4.13 : Çift entegratör sistemi için Kalman filtresi uygulaması (eleman sayısı :20).....	61
Şekil 4.14 : Çift entegratör sistemi için Kalman filtresi uygulaması (eleman sayısı :10).....	61
Şekil 4.15 : Sistemdeki elemanların bilgilerinin oluşturulması.	63
Şekil 4.16 : Sistemdeki elemanların komşuluk ilişkisinin oluşturulması	63
Şekil 4.17 : Sistemde bulunan elemanlar için kontrol girdilerinin oluşturulması	64
Şekil 4.18 : Sisteme gürültülerin eklenmesi.	65
Şekil 4.19 : Kalman filtresi uygulanması.....	66
Şekil 4.20 : Harmonik osilator sistemi için gürültüsüz ortam (eleman sayısı: 30)... ..	66
Şekil 4.21 : Harmonik osilator sistemi için gürültüsüz ortam (eleman sayısı: 20)... ..	67
Şekil 4.22 : Harmonik osilator sistemi için gürültüsüz ortam (eleman sayısı: 10)... ..	67
Şekil 4.23 : Harmonik osilator sistemi için gürültülü ortam (eleman sayısı: 30).	67
Şekil 4.24 : Harmonik osilator sistemi için gürültülü ortam (eleman sayısı: 20).	68
Şekil 4.25 : Harmonik osilator sistemi için gürültülü ortam (eleman sayısı: 10).	68
Şekil 4.26 : Harmonik osilator için Kalman filtresi uygulaması (eleman sayısı:30). ..	69
Şekil 4.27 : Harmonik osilator için Kalman filtresi uygulaması (eleman sayısı:20). ..	69
Şekil 4.28 : Harmonik osilator için Kalman filtresi uygulaması (eleman sayısı:10). ..	69
Şekil 4.29 : Sistemdeki elemanların bilgilerinin oluşturulması.	70
Şekil 4.30 : Sistemdeki elemanların komşuluk ilişkisinin oluşturulması	71
Şekil 4.31 : Sistemde bulunan elemanlar için kontrol girdilerinin oluşturulması	71
Şekil 4.32 : Sisteme gürültülerin eklenmesi.	72
Şekil 4.33 : Kalman filtresi uygulanması.....	72
Şekil 4.34 : Sinüzoidal sistem için gürültüsüz ortam (eleman sayısı: 30)	73
Şekil 4.35 : Sinüzoidal sistem için gürültüsüz ortam (eleman sayısı: 20).	73
Şekil 4.36 : Sinüzoidal sistem için gürültüsüz ortam (eleman sayısı: 10).	74
Şekil 4.37 : Sinüzoidal sistem için gürültülü ortam (eleman sayısı: 30).	74
Şekil 4.38 : Sinüzoidal sistem için gürültülü ortam (eleman sayısı: 20)	74
Şekil 4.39 : Sinüzoidal sistem için gürültülü ortam (eleman sayısı: 10).	75
Şekil 4.40 : Sinüzoidal sistem için Kalman filtresi uygulaması (eleman sayısı: 30). ..	75
Şekil 4.41 : Sinüzoidal sistem için Kalman filtresi uygulaması (eleman sayısı: 20). ..	76
Şekil 4.42 : Sinüzoidal sistem için Kalman filtresi uygulaması (eleman sayısı: 10). ..	76

ÇOK ETMENLİ SİSTEMLERİN İŞ BİRLİĞİ KONTROLÜNDE KALMAN FİLTRESİ UYGULAMALARI

ÖZET

Son yıllarda kendi başına karar veren ve bu kararı uygulayan sistem veya araçlara olan ilgi giderek artmaya başlamıştır. Özellikle gerçekleşen Endüstri 4.0 devrimiyle birlikte üretimde verimliliğin artırılması, fabrika içi organizasyonun iyileştirilmesi ve işçi sağlığını riske atan çoğu uygulama yerini robotik veya insansız çözümlere bırakmıştır. Çok sayıda eleman veya araçtan oluşan bu robotik veya insansız sistemlere çok etmenli sistem adı verilmektedir.

Otonom sistemlere olan ilginin artması, bu sistemleri oluşturan elemanların sahip oldukları parametrelerin (hız, konum vb.) bilinmesi ve anlık olarak kontrol edilmesi ihtiyacını da artırmaktadır. Sistemdeki elemanların tek tek bilgilerinin alınması ve bu bilgilere göre bu elemanların kontrolü, özellikle çok sayıda elemandan oluşan sistemler için oldukça zahmetlidir. Bu problemi çözmeye adına ağ topolojisine sahip çok etmenli sistemler kullanılmaktadır. Söz konusu sistemlerle ilgili detaylı bilgi verilmiştir.

Yapılan tez çalışmasında ağ topolojisine sahip çok etmenli sistemlerin belirli özellikleri hakkında bilgi verilmiştir. Çok etmenli sistemlerin bireysel elemanlardan oluşan sistemlere göre olumlu ve olumsuz yönleri aktarılmıştır.

Çok etmenli sistemlerdeki elemanlar arasındaki haberleşme belirli bir ağ topolojisine göre gerçekleşmektedir. Bu topolojiyi görsel olarak da açıklayan graf teorisi hakkında detaylı bilgi verilmiştir. Graf teorisine göre, çok etmenli sistemlerdeki elemanların iş birliğine varıp beraber hareket edebilmeleri için hangi koşulların sağlanması gerektiği hakkında detaylı bilgi verilmiştir.

Yapılan tez çalışmasında 3 farklı çok etmenli sistem ele alınmıştır. Çalışmanın amacı, söz konusu sistemlerdeki elemanların sahip oldukları bilgilere gürültü eklenmesine rağmen, söz konusu elemanların yine de düzgün bir şekilde iş birliğine varmalarını sağlamaktır. Bunu sağlayabilmek için Kalman filtresi uygulamaları gerçekleştirilmiştir. Bu sayede, bozucu girdilere rağmen sistemdeki elemanlar düzgün bir şekilde iş birliğine varmıştır.

Yapılan uygulamalarda, ele alınan sistemlerdeki elemanların sahip oldukları bilgilerin başlangıç değerleri rastlantısal bir şekilde oluşturulmuştur. Sistemdeki elemanların komşuluk ilişkilerinin oluşturulabilmesi için, söz konusu elemanlar belirli bir düzlemde rastgele olacak şekilde konumlandırılmıştır. Sistemdeki elemanlar aralarındaki mesafeye göre komşu olarak değerlendirilmiştir.

Sistemlerde bulunan elemanlar için iş birliği protokolüne bağlı olarak, adaptif PD ve gradyant düşümü yöntemlerinden de yararlanılarak kontrol girdileri oluşturulmuştur.

Sistemlere normal dağılıma bağlı olarak ölçüm ve proses gürültüleri dahil edilmiştir. Sistemlerdeki elemanların arasındaki bilgi transferini etkileyen bu bozucu girdilere

rağmen, sistemin dinamiğine bağı olacak biçimde Kalman filtresi uygulaması gerçekleştirilerek sistemdeki elemanların düzgün bir biçimde iş birliğine varmaları sağlanmıştır. İlgili sonuçlar grafiksel olarak da gösterilmiştir.



CONSENSUS OF MULTIAGENT SYSTEMS WITH KALMAN FILTER APPLICATIONS

SUMMARY

In recent years, interest in systems or tools that can make their own decisions and implement these decisions has begun to grow. Especially with the realization of the Industrial 4.0 revolution, many applications that raise productivity in production, improve in-plant organization and raise workers' health are left to robotic or unmanned solutions. These robotic or unmanned systems consisting of a large number of elements or vehicles are called multi-agent systems in the technical literature.

It has been shown in this study that these multi-agent systems, which are formed by bringing more than one vehicle or element according to the neighborhood relation, are more advantageous than the systems consisting of the elements which are independent of each other and which decide independently.

Increased interest in autonomous systems has also increased the need to know and instantly control the parameters (speed, location, etc.) of the elements that make up these systems.

Taking individual information of the elements in the system and controlling them according to this information is especially troublesome for systems consisting of a large number of elements. In order to solve this problem, multi-agent systems with network topology will be discussed.

In the thesis study, detailed information about specific properties of multi-agent systems with network topology is provided. According to this, the elements forming these systems have network topology and share the information they have with the elements forming the system with the help of various relation rules.

In the thesis study, this neighborhood relation is emphasized and detailed information about information transfer topology is provided

The members of the multi-agent systems have to work together to accomplish a certain task or achieve a goal. Detailed information about the meaning and conditions of consensus and cooperation is provided in this study. In order to be able to cooperate in such systems, a specific consensus rule or protocol has to be established.

In the thesis study, it has been shown how consensus and cooperation are provided according to the parameters and equations in systems with different dynamics. These rules, which allow the elements in the system to agree on a common value and act accordingly, are all called the consensus algorithm.

The definition of consensus and conditions under which consensus is achieved is given in detail. Detailed information about graph theory and consensus conditions under the graph theory is provided.

Consensus of multi-agent systems is analyzed with respect to graph theoretic concepts. Sample problems and solutions have been included in order to better show how and under what conditions a specific system reaches consensus.

In the study conducted, the advantages and disadvantages of the multi-agent system components according to the individual studies of the collective coordination and working together are also revealed. It has been observed that the performance of multi-agent systems based on the consensus algorithm is higher.

In the thesis study, it is explained how the elements of multi-agent system structure share the information they have with other members. It is analyzed how individual agents behave when noise is added to the communication links between them.

Despite the noise in the system, Kalman filter applications have been implemented in order to ensure that the elements in the system are nevertheless neatly connected to the system dynamics.

The main work in this thesis is to investigate the effect of noise (process and measurement noise) in consensus and how to suppress this noise using Kalman filter. Three different systems are analyzed to ensure the consensus of these multi-agent systems are obtained under noisy conditions. Detailed information and equations is given to know the dynamics and consensus conditions for these multi-agent systems.

In order to better explain the theory of the applied Kalman filter, information has been given about probability and distributions. In particular, detailed information has been provided about the normal distribution which will be evaluated as a disturbing input. Therefore, it is shown how the estimates depend on the parameters. At the same time, parameters of how the linear state estimation methods are derived and connected are explained. Detailed informations about the parameters used in Kalman filter equations are given. The operation diagram of the Kalman filter algorithm is described.

In case the system is nonlinear, standard Kalman filter cannot be used. In this case extended Kalman filter can be employed to estimate system states.

In the thesis, all these information data are explained in detail with multi-agent systems with 3 different dynamics and different parameters in various directions.

The equations and parameters of the mentioned systems are explained. The information transfer topology and applied consensus protocols for these systems are described in detail. Noise is added to the information exchanged between these elements and the transfer of this information, and Kalman filter applications are performed to suppress this noise.

The obtained results are illustrated by graphs. The values of the number of agents are changed and the results are given comparatively.

The elements in the multi-agent systems will share the information they have by communicating within a specific neighborhood relationship. The probability of systems being able to communicate, that is, their neighbors, will be determined randomly, depending on their proximity.

Each system considered has a specific control protocol. The parameters required for this control protocol are set iteratively by means of adaptive PD controller which is tuned by the gradient descent algorithm.

It is assumed that the information exchange between agent is noisy and therefore, the systems are simulated by adding noise to these informations. Consequently, signals to observe the effects of this noise, Kalman filter is applied to suppress this noise, so that smoother information is transmitted to the elements forming the multi-agent.





1. GİRİŞ

Son yıllarda otonom araç veya elemanlardan oluşan sistemlerde görülen artış, buna bağlı olarak kontrol algoritmalarının daha fazla iyileştirilmesi durumunu ve mecburiyetini gündeme getirmiştir.

Belirli bir görevi tek başına ve kendi karar vererek yerine getiren otonom araçlara göre, birlikte çalışan ve birbirleriyle bilgi paylaşımı yaparak aynı görevi yerine getiren otonom araçların daha yüksek verimde ve hızda olduğu görülmüştür.

Çok etmeden oluşan otonom sistem veya araçların ekip olarak çalışabilmeleri için, belirli bir bilgi veya değerde anlaşmaları gerekmektedir. Bu yapıdaki elemanların iş birliğine ulaşip beraberce çalışabilmeleri için algoritmik yöntemler geliştirilmiştir. Bu algoritma yapılarındaki temel amaç, komşuluk ilkesine dayalı olarak, çok etmenli sistemdeki araç veya elemanların birbiriyle bilgi paylaşımında bulunup ortak bir değerde anlaşabilmeleri ve beraber hareket edebilmeleridir.

Bir ağ yapısındaki veya çok etmenli bir sistemdeki iş birliği, sistemi oluşturan bütün elemanların durumlarına bağlı olan belirli bir değerde anlaşmaya varmaları anlamına gelmektedir. İş birliği algoritması ise, bu anlaşmayı çerçeveleyen ilişki kurallarını düzenler [1]. Buna göre, ağ yapısındaki her bir elemanın komşularıyla veya diğer elemanlarla olan bilgi alış verişi düzenlenir.

Çok etmenli bir sistemdeki elemanların birlikte ve koordine bir şekilde çalışması, otonom çalışan sistemlere ilginin artmasına paralel bir şekilde, bir çok araştırmacının ilgisini giderek artan bir yönde bu alana çekmiştir. Çok etmenli sistem yapılarına gösterilen ilgideki bu artış, bu yapıyı oluşturan elemanların da bir iş birliği içerisinde olmalarının önemini göstermiştir. İş birliği için ilk olarak, ağ yapısını oluşturan elemanların veya araçların davranışlarından bağımsız veya davranışlarına bağlı olarak, ortak bir değere ulaşmaları gerekmektedir [2].

Çok etmenli sistem yapısındaki elemanlar belirli bir değişken değerinde veya durum bilgisinde anlaştıkları zaman, bu, iş birliğine vardıkları anlamına gelir. Belirli bir bilgide iş birliğine varılması, koordinasyon görevi bakımından kritik olan bilginin

komşuluk yapısı üzerinden elemanlarca paylaşılmasını garanti altına alır. İş birliğine ulaşmak için, sistemdeki elemanların durum bilgilerinin uygun iş birliği algoritması belirlenerek paylaşılması gerekir. Söz konusu durum bilgisi koordinasyon için gerekli olan değişken değerinin bir örnekleme olarak düşünülebilir. İş birliği algoritmalarında bilgi, ağ yapısındaki elemanların komşuluk ilişkilerine göre paylaşılır [3]. Çok etmenli sistem yapısını oluşturan her bir eleman kendi komşusuna göre durum bilgisini günceller. Burada amaç, ilgili sistemdeki her bir elemanın durum bilgilerinin ortak bir değere yakınsadığı bir kural tasarlamaktır.

1.1 Amaç ve Kapsam

Çok etmenli bir sistemde, sisteme ait elemanların veya araçların birbirleriyle uyumlu ve daha da önemlisi iş birliği halinde çalışması için ana etken haberleşmenin düzgün bir biçimde yapılabilmesidir.

İş birliği algoritmalarında komşuluk ilkesi esastır. Bu komşuluk ilkesine göre, birbiriyle komşu olan iki eleman veya araç, taşıdıkları konum, pozisyon, hız vb. değişkenler hakkında bilgi paylaşımı yaparlar. Yapılan çalışma ile bu bilgi paylaşımlarındaki bozucu girdi etkilerinin Kalman filtresi uygulamaları yardımıyla önüne geçilmesi amaçlanmaktadır.

İlgili tez çalışması, belirli bir çok etmenli sistemde, birbirleriyle haberleşmeye çalışan araç veya elemanların, bozucu girdiye rağmen Kalman filtresinin uygulanması sayesinde ve iş birliği algoritması çerçevesinde ortak bir değerde anlaşabilmelerini kapsamaktadır.

1.2 Çalışmanın Ana Hatları

Tezde yapılan çalışmalar belirli bir sıralamaya göre anlatılacaktır. Bu sıralamaya göre tez çalışmasının çıktısı olan algoritmanın barındırdığı teori ve yöntemler hakkında bilgilendirme yapılması ve bu sayede çalışmanın daha iyi aktarılması hedeflenmektedir.

Çalışmanın ikinci bölümünde genel olarak ağ yapısı ve ağ yapısını oluşturan elemanların iş birliği hakkında bilgi verilecektir.

Çalışmanın üçüncü bölümünde iş birliği algoritmasının maruz kaldığı bozucu girdilere karşı kullanılan Kalman filtresi hakkında temel bilgiler verilip, genişletilmiş Kalman filtresi hakkında bilgi verilecektir.

Çalışmanın dördüncü bölümünde tezin çıktısı olan algoritma ele alınıp, algoritmada kullanılan parametre ve denklemler hakkında detaylı bilgi verilecektir. Aynı zamanda algoritmanın işleyişi kapsamlı bir şekilde anlatılacaktır.

Çalışmanın son bölümünde ise sonuçlar ve ileride yapılması hedeflenen çalışmalar aktarılacaktır. Sonuçlar bir tablo halinde verilecektir.





2. ÇOK ETMENLİ SİSTEMLERDE İŞ BİRLİĞİ

2.1 Çok Etmenli Sistem Yapısında İş Birliğinin Anlamı

Çok etmenli sistem yapısında iş birliğini anlatabilmek için öncelikle bu yapıdaki elemanlar ve bu elemanların bir araya gelerek oluşturduğu sistemler hakkında bilgi verilecektir.

Çok etmenli bir sistem yapısını oluşturan elemanlar (agents), çevreyi çeşitli algılayıcılar yardımıyla algılayan ve buna göre hareket geliştiren yapılar şeklinde tanımlanabilir [4].

Çok etmenli sistemin içerisinde yer alan herhangi bir eleman, dinamik bir ortamda yaşayıp, özerk bir şekilde hareket eden ve kendisine atanan bir dizi görev veya hedefi yerine getiren yapılar olarak tanımlanabilir [5].

Çok etmenli sistem yapısı, bünyesinde bulundurduğu elemanların bireysel olarak çözemediği zorluklara karşı bir araya gelerek birlikte çalıştığı yapıdır [6].

Çok etmenli sistem yapıları için; insansız taşıtlar, birden fazla robottan oluşan sistemler, iletişim ağları vb. örnek olarak verilebilir [7]. İş birliğinin sağlanması çok etmenli bir sistem için oldukça önemlidir. Çok etmenli sistemdeki elemanların birlikte çalışması için uygun bir iş birliği protokolü ve algoritması gereklidir.

Çok etmenli sistemdeki elemanların veya araçların birlikte çalışması sivil ve askeri alanda yaygın olarak kullanılmaktadır. Koordine olarak uyumlu çalışan araçlar, bütün görevi tek başına getiren bir araca göre mühendislik açısından daha fonksiyoneldir.

Çok etmenli bir sistemdeki elemanlar ağ yapısına benzer şekilde birbirlerine bağlıdır. Bu ağ yapısı için, ağı oluşturan elemanların koordinasyonu, bu elemanların bazılarının veya hepsinin bilgiyi paylaşarak ortak bir hedef belirlemeleri anlamına gelmektedir.

Çok etmenli bir sistemdeki elemanların, iş birliğini sağlayarak koordine çalışması için aşağıda belirtilen şartları yerine getirmesi gerekir [8]:

- Sistemi oluşturan elemanlar için ortak bir hedefe yönelik ortak bir davranış gerekir.
- Sistemi oluşturan elemanların birbirleriyle iletişim kurması gerekir.
- Sistemi oluşturan elemanların, oluşturdukları sistemin performansını artırması veya sistem için zamandan kazanç sağlaması gibi süreci iyileştirici unsurlar için bir araya gelmesi gerekmektedir.

Çok etmenli bir yapıdaki elemanların koordinasyon içinde çalıştığı bir sistemi tasarlarken, aşağıda belirtilen zorluklarla karşılaşmaktadır [8]:

- Tek bir sistemi tasarlamak yerine bütün bir sistemi alt sistemlere bölerek tasarlama gereği duyulur.
- Sistemi oluşturan elemanların birbirleriyle olan iletişimleri genellikle sınırlı ve gürültülüdür.
- Bütün bir sistemin karmaşık olması ve sisteme yeni elemanların eklenmesi sistemin lineerliğini korumasını zorlaştırır. Zamanla lineer olmayan bir sisteme dönüşebilir.
- Sistem yapısında bulunan araçların bireysel hedefleriyle takım hedefleri arasında karar verme süreci hemen hemen aynı anda gerçekleşir.

İş birliği halinde çalışan araçların oluşturduğu çok etmenli sistemlerin, her bir aracın tek başına ve bireysel olarak görevleri yerine getirdiği sistemlere göre avantajları aşağıda belirtilmiştir [8]:

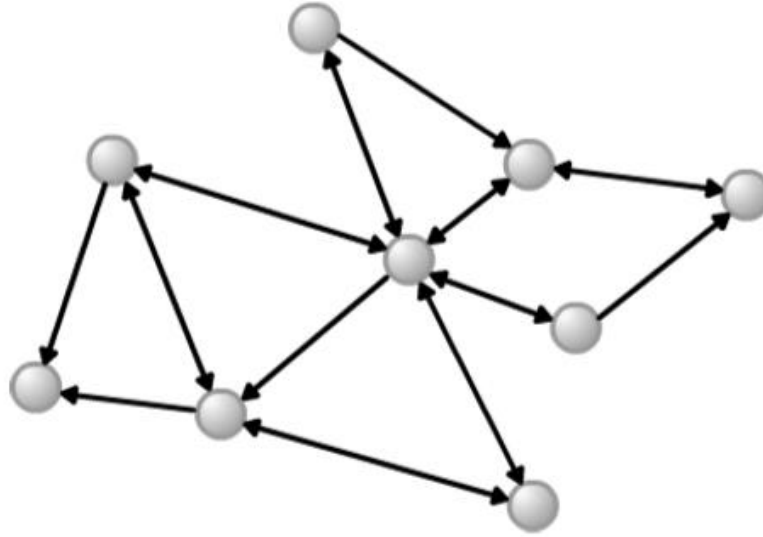
- Bireysel araçlardan oluşan sistemlere göre daha iyi hata toleransı mevcuttur.
- Bireysel araçlardan oluşan sistemlere göre daha karmaşık veya zor olan görevler yerine getirebilir.
- Çevresel etkenlere karşı daha hızlı reaksiyon gösterir.

2.2 Graf Teorisi

Bu bölümde çok etmenli sistem yapısındaki elemanların durum bilgilerinin belirli bir ortak değerde anlaşmaları için kullanılan graf teorisi hakkında bilgilendirme yapılacaktır.

Gerçek dünyada karşılaşılan çok etmenli sistem yapısındaki çoğu uygulama, iki noktanın birleştiği çizgilerden oluşan diyagramlarla tanımlanabilir [9]. Bu mantık, ağdaki elemanların iletişim dinamiklerini gösteren graf teorisiyle doğrulanabilir.

Belirli bir bağlantı yoluyla birbirleri arasında bilgi paylaşan çok etmenli sistem yapısındaki elemanları inceleyelim. Elemanlar arasındaki bağlantı, bazılarında tek yönlü, bazılarında ise iki yönlü olabilir. Yani paylaşılan bilgi bazılarında tek bir yönde iken bazılarında ise iki yönde de olacak şekildedir. Şekil 2.1’de belirtilen oklar bilgi transferini temsil eder [10]. Şekil 2.1’deki bazı bağlantılar bilgiyi tek bir yönde, bazıları ise iki yönde de göndermektedir. Bilgi aktarım yönünü gösteren oklar da dahil olmak üzere, bu bağlantı düzeni bilgi aktarım topolojisi olarak bilinir.



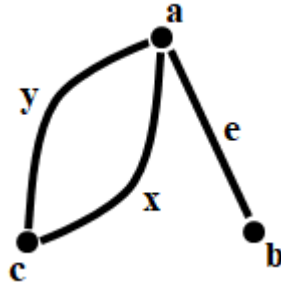
Şekil 2.1 : Bir graf topolojisi [10].

Bir graf yapısı, $N = \{1, 2, \dots, n\}$ ve $E \subseteq N \times N$ olan bir $G = (N, E)$ şeklinde gösterilir [10]. Graf teorisine göre, çok etmenli sistem yapısını oluşturan her bir eleman bir düğüm olarak tanımlanır. n adet eleman varsa n adet de düğüm bulunur. $V = \{v_i : i \in N\}$ kümesi bütün v_i düğümlerini içerir.

E ise düğümler arasındaki bağlantılar kümesidir ve G grafinin kenarlarını temsil eder. Yani E , j düğümünün bilgiyi i düğümünden aldığını gösteren bütün (i, j) kümeleridir. Düğümlerin komşu düğümlerini gösteren N_i , $N_i = \{j \in N : (j, i) \in E\}$ şeklinde tanımlanır.

Belirli bir grafta her $(i, j) \in E$ için, (j, i) oluyorsa bu grafa yönsüz graf (undirected graph) denir [10]. Bu durum, çok etmenli sistem yapısını oluşturan elemanların iş birliği konusu düşünülecek olursa, sistemdeki her iki eleman arasında bilgi alışverişinin, her iki yönde de belirli bir bağlantı üzerinden gerçekleştiğini gösterir.

Yönsüz graf için iletişim yönü belirtilmez. Aşağıdaki Şekil 2.2’de belirtildiği gibi e kenarı $e = \{a, b\} = \{b, a\}$ şeklinde tanımlanabilir.



Şekil 2.2 : Yönsüz (undirected) graf.

Eğer Şekil 2.1’deki bütün bağlantı okları iki yönlü olsaydı, bu grafi bir yönsüz graf olarak tanımlayabilirdik. Eğer bir graf yönsüz değilse, yönlü graf (directed graph) veya digraph olarak adlandırılır. Şekil 2.1’de gösterilen graf yönlü bir graftır. Eğer yönsüz bir grafta herhangi iki düğüm noktasını birbirine bağlayan bir yol varsa söz konusu graf bağlı graf olarak adlandırılır. Eğer yönlü bir grafta, herhangi iki düğüm noktasını birbirine bağlayan bir yol varsa söz konusu graf güçlü-bağlı graf olarak adlandırılır [10].

Grafların alt kümeleri alt grafları oluşturur. Herhangi bir grafın bütün düğüm noktalarını kapsayan alt grafına tarama ağacı (spanning tree) adı verilir [10].

Graf teorisinde bilginin düğümler arasında nasıl transfer edildiği açıklanırken iki önemli matris kullanılır. Bunlardan ilki komşuluk matrisi olarak adlandırılır. Komşuluk matrisi de ağırlıklı (weighted) ve ağırlıksız (unweighted) olmak üzere ikiye ayrılır.

“Unweighted” yani ağırlıksız komşuluk matrisine göre, komşuluk matrisinin elemanları aşağıdaki şartları sağlar [10]:

$$a_{ij} = \begin{cases} \text{eğer } (j, i) \in E \text{ ise} & 1 \\ \text{aksi takdirde} & 0 \end{cases} \quad (2.1)$$

“Weighted” yani ağırlıklı komşuluk matrisine göre, komşuluk matrisinin elemanları aşağıdaki şartları sağlar [10]:

$$a_{ij} = \begin{cases} \text{eğer } (j, i) \in E \text{ ise } \sigma_{ij} > 0 \\ \text{aksi takdirde} & 0 \end{cases} \quad (2.2)$$

Bu tanımlardan yola çıkarak, eğer G grafi yönsüz ise; A komşuluk matrisi simetriktir ve $A = A^T$ diyebiliriz.

Graf teorisinde bir diğer önemli matris ise Laplacian matrisidir. $L = [l_{ij}]_{i,j=1}^n$ Laplacian matrisinin elemanları aşağıdaki şekilde tanımlanır [3]:

$$l_{ij} = \begin{cases} \sum_{j \neq i} a_{ij} & i = j \\ -a_{ij} & i \neq j \end{cases} \quad (2.3)$$

a_{ij} , komşuluk matrisinin elemanlarıdır. Bu tanımdan yola çıkarak; eğer G grafi yönsüz ise, L simetriktir ve $L = L^T$ diyebiliriz.

$G(V, E, A)$ derecesi $n \geq 2$ olan bir yönsüz (undirected) graf olsun. $V = \{v_1, v_2, v_3 \dots v_n\}$ düğümler kümesi ve $E \subseteq V \times V$ kenarların kümesi olur. Aynı zamanda $A = [a_{ij}] \in R^{n \times n}$ komşuluk matrisini oluşturur. Yönsüz bir grafta, $e_{ij} = e_{ji}$ olur. Sadece ve sadece $a_{ij} > 0$ olduğu durumda $e_{ij} \in E$ diyebiliriz. Bir yönsüz graf, her zaman bağlıdır [3].

Bir v_i düğümünün dış-derecesi $deg_{out}(v_i) = \sum_{j=1}^n a_{ij}$ şeklinde tanımlanır [3]. G graf matrisinin derecesi $D = [d_{ij}]$ şeklinde diyagonal bir matristir. Bütün $i \neq j$ değerleri için $d_{ij} = 0$ ve $d_{ii} = deg_{out}(v_i)$ olur. Graf ile ilgili olarak Laplacian matrisi aşağıdaki şekilde tanımlanır [3]:

$$L = [l_{ij}] = D - A = \begin{cases} \sum_{j=1, j \neq i}^n a_{ij}, & j = i, \\ -a_{ij}, & j \neq i. \end{cases} \quad (2.4)$$

Herhangi bir $i, j \in E$ ve $i \neq j$ için, i 'den j 'ye giden bir yol varsa söz konusu graf, bağlı-graf (connected graph) olarak adlandırılır [11]. Çok etmenli bir sistem yapısı için, G grafındaki her bir eleman bir i düğümü ile temsil edilir. Her bir (i, j) kenarı ise j elemanından i elemanına doğru olan iletişim kanalını temsil eder.

Yönsüz (undirected) bir iletişim grafına sahip çok etmenli sistemler için iş birliğine ulaşma durumu aşağıda incelenmiştir. Çok etmenli sistem yapısındaki her bir eleman yukarıda belirtilen L Laplacian matris değerine sahiptir ve bu Laplacian matrislerin özdeğerleri aşağıdaki gibi sıralanabilir [12]:

$$0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N \quad (2.5)$$

Eğer yönsüz bir graf bağlı ise $\lambda_2 > 0$ elde edilir.

Yönsüz bir iletişim grafına sahip bir çok etmenli sistem, 2.7 denkleminde verilen kontrol protokolü uygulandığında 2.6 denkleminde verilen koşulu sağlıyorsa, söz konusu ağ yapısındaki elemanlar için iş birliğine ulaşılır diyebiliriz [12].

$$\lim_{k \rightarrow \infty} \|x_i(k) - x_j(k)\| = 0 \quad \forall i, j \in N \quad (2.6)$$

$$u_i(k) = K \sum_{j=1}^N a_{i,j} [x_j(k) - x_i(k)], \quad i \in N, \quad K \in R^{m \times n} \quad (2.7)$$

Yönlü (directed) bir iletişim grafına sahip çok etmenli sistemler için iş birliğine ulaşma durumu ise aşağıdaki gibidir. $\dot{x} = Bx(t)$ denkleminde sahip bir sistemi düşünelim. B matrisi için özdeğerleri kümesi $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ olsun. Ağ yapısındaki elemanlar aşağıdaki koşulları sağlarsa iş birliğine ulaşılabilir [13]:

- B matrisinin bütün satırlarının ayrı ayrı toplamı sıfıra eşit olması gerekir.
- $Re(\lambda_i) > 0, \forall \lambda_i \in \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ koşulunun sağlanması gerekir.

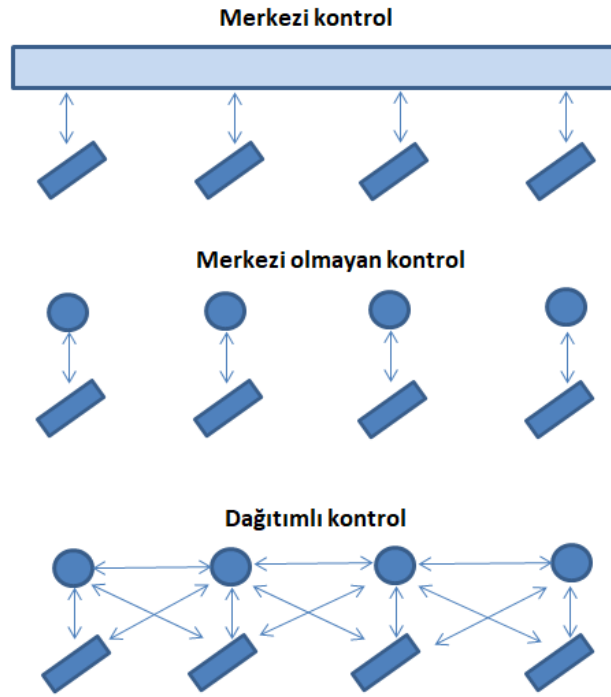
2.3 İş Birliği Algoritmasının Temelleri

Bu bölümde çok etmenli sistem yapısındaki elemanların iş birliği için geliştirilen iş birliği algoritmasıyla ilgili teorik altyapı anlatılacaktır.

Kontrol sistemlerinde, dinamik bir sisteme sahip çok etmenli sistem yapısı, bünyesindeki bütün elemanlar diğer bütün elemanların durum bilgilerine bağlı belirli

bir değere yakınsıyorsa bu sistem yapısında iş birliği sağlanabilir denebilir. Çok sayıda elemandan oluşan bu ağ yapılarını oluşturan elemanların ortak bir değere yakınsayabilmeleri için bir iş birliği protokolü veya algoritması oluşturulması gerekmektedir. İş birliği algoritmaları elemanların komşuları arasındaki bilgi alış verişine bağlıdır [14]. Belirli bir dinamiğe sahip elemanlar, belirli bir iletişim topolojisi ve komşuluk ilişkileri yardımıyla durumlarını günceller ve iş birliği sağlanır.

Çok etmenli sistemler için üç çeşit kontrol topolojisi mevcuttur. Bu topolojiler sisteme ait elemanları kontrol edebilmek için; merkezi kontrol (centralized controller), merkezi olmayan kontrol (decentralized controller) ve dağıtılmış kontrol (distributed controller) olmak üzere sıralanabilir [15].



Şekil 2.3 : Kontrol yöntemleri.

Bir dinamik sistemi veya ağ yapısında bulunan otonom araçları düşündüğümüzde, iş birliğine ulaşmanın anlamı, sistemdeki veya ağdaki bütün eleman veya araçların durum bilgilerine bağlı olarak belirli bir değerde anlaşmalarıdır. İş birliği algoritması ise ağ yapısında bulunan bir elemanın komşularıyla olan haberleşmesini belirleyen ilişki kurallarını oluşturur.

Örnek olarak bir yangın durumunu ve bu yangını söndürmekle yükümlü otonom olarak çalışan robot itfaiyeleri düşünelim. Her bir robot, yangının çapı, çevrenin çapı

gibi koordinasyon parametrelerini belirler ve bu deęerleri rnekendirerek yakınındaki dięer robotlarla bu bilgileri paylařır. Yangınla mcadelenin etkili bir řekilde yapılabilmesi iin, her bir robotun kendisini yangın merkezinden belirli bir uzaklıkta konumlandırarak sndrme iřlemini gerekleřtirmesi gerekmektedir. Bu yzden, yangın sndrmekle ykml bu robotların yangının etkin bir řekilde sndrlmesi iin gereken ap zerinde ve aralarında belirli bir mesafe olacak řekilde dizilmeleri iin bir anlařmaya varmaları gerekmektedir. ok etmenli sistem yapısında alıřan bu otonom robotlar, bozuk sensrlere, kesilen haberleřmeye, grltl baęlantılara raęmen iletiřimi srdrerek anlařmaya varmak durumundadır.

ok etmenli sistem yapısındaki elemanların durum bilgileri zerinde iř birlięine varmaları iin belirlenen iř birlięi algoritmasında, her bir eleman durum bilgisini, grltl ve zamanla deęiřen bir ortamda dięer elemanlarla paylařarak gnceller. Yani her bir ara veya eleman durum bilgisini komřularının durum bilgilerine gre gncel tutmaya alıřır. Bu yzden iř birlięi algoritması, sistemdeki btn elemanların durum bilgilerinin ortak bir deęere yakınsadıęı ve bu sayede btn elemanların neredeyse benzer dinamiklere sahip olduęu bir gncelleme kuralları btndr. Eęer haberleřme topolojisi srekli ise, her bir aracın durum bilgisini gncellemesi diferansiyel bir denklem ile modellenir. Fakat iletiřim topolojisi kesikli ise, gncelleme kuralı fark denkleminde gre belirlenir [8].

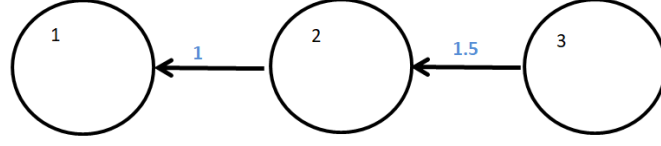
Yaygın ve temel olarak kullanılan iř birlięi algoritması ařaęıdaki řekildedir [8]:

$$\dot{x}_i(t) = - \sum_{j=1}^n a_{ij}(t)(x_i(t) - x_j(t)), \quad i = 1, 2, \dots, n \quad (2.8)$$

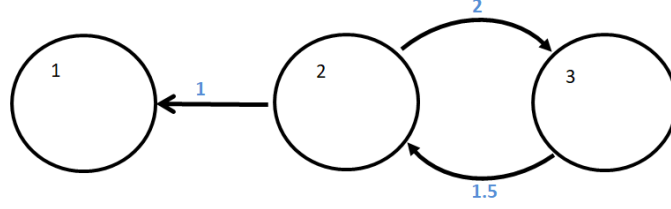
Bu denklemdeki $a_{ij}(t)$, t anındaki bir grafta (i, j) komřuluk matrisidir. $x_i(t)$ ise i 'ninci aracın durum bilgisini gsterir.

$a_{ij} = 0$ olduęu durumda, bu j aracının i aracından bilgi alamadıęını gsterir. stteki denklemin bir sonucu olarak, i aracının $x_i(t)$ durum bilgisinin, komřularının durum bilgilerine doęru srklendięi anlařılır. İlgili denklem, araçların bilgi durumlarının uzlařtıęını gsterse de belirli bir ortak deęeri dayatmaz.

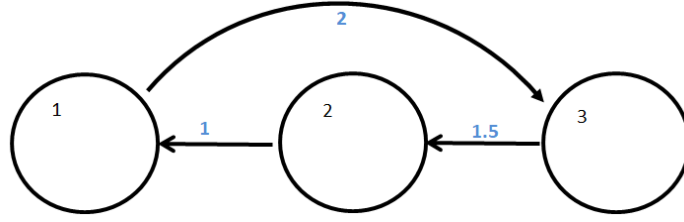
Ařaęıdaki řekiller rnek olarak 3 farklı aracın 3 farklı iletiřim sahip olduęu durumları aıklamaktadır [8].



(a)



(b)



(c)

Şekil 2.4 : Üç farklı graf için iletişim topolojileri. (a) ve (b) güçlü-bağlı değildir, (c) güçlü-bağlıdır [8].

Bu kısımda, iletişim topolojisinin zamana bağlı olduğu ve a_{ij} değerlerinin sabit olduğu yani simetrik olmayan Laplacian matrisi L_n 'nin sabit olduğu durumda, iş birliği algoritmasındaki elemanların yakınsadığı koşulları inceleyeceğiz.

Yukarıda belirtilen üç farklı iletişim topolojisine sahip graf için Laplacian matrisleri sırasıyla aşağıdaki gibi olur.

$$L_1 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1.5 & -1.5 \\ 0 & 0 & 0 \end{bmatrix}$$

$$L_1 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1.5 & -1.5 \\ 0 & -2 & 2 \end{bmatrix}$$

$$L_1 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1.5 & -1.5 \\ -2 & 0 & 2 \end{bmatrix}$$

Yakınsamaya ulaşmak için, L_n 'nin özdeğerinin sıfır olduğuna emin olunması gerekmektedir. Yukarıda belirtilen bütün Laplacian matrislerinin sıfır özdeğeri olmasına rağmen, Şekil 3.1(a) ve Şekil 3.1(b)'dekiler güçlü-bağlı değil, Şekil

3.1(c)'deki ise güçlü-bağılıdır. Yukardaki matrislerin tek ortak özelliği, L_1 , L_2 ve L_3 yönlü graflarının hepsinin de yönlü tarama ağacı (spanning tree) kökenli olmasıdır. Daha öncede belirtildiği gibi, L_n 'nin özdeğerinin sıfır olması, sadece ve sadece ilgili yönlü grafin yönlü bir tarama ağacına sahip olması durumuna bağlıdır. Bu sonuçtan yola çıkarak, denklem 2.8'in, yönlü graf topolojisi için yönlü bir tarama ağacına sahip olması durumunda, yönsüz bir graf topolojisi için de güçlü bağlı olduğu durumda, iş birliğine ulaşacağını söyleyebiliriz.

Çok etmenli sistem yapısındaki elemanlar, belirli bir değişken üzerinde anlaşmaya vardıkları zaman, iş birliğine ulaştıkları sonucunu çıkarabiliriz. Durum bilgilerinde anlaşmaya varılması, belirli bir haberleşme topolojisi üzerinden bilgi paylaşan araçların, koordinasyon görevi için gerekli olan bilgilere tutarlı bir şekilde ulaştıkları anlamına gelir. Bu durumda, iş birliği elde etmek için, durum bilgileri olarak adlandırılan ortak bir değişken olmalı ve değişken üzerinde anlaşmaya varmak için uygun algoritmik yöntemler uygulanmalıdır. Durum bilgisi, ağ yapısındaki araçların veya elemanların koordine olması için gerekli olan değişkene bir örnek olarak verilebilir.

x_i , i 'ninci elemanın durum bilgisi olsun. Durum bilgisi, iş birliği içindeki çok etmenli sistem elemanları arasında gerekli olan bilgiyi temsil eder. Sistem yapısındaki elemanlar açısından durum bilgisi, konum, hız, ivme gibi değişkenler olabilir. Yukarıda verilen 2.5 denklemi aşağıdaki şekilde de yazılabilir [16]. Bu denklem sürekli-zaman (continious-time) için geçerlidir:

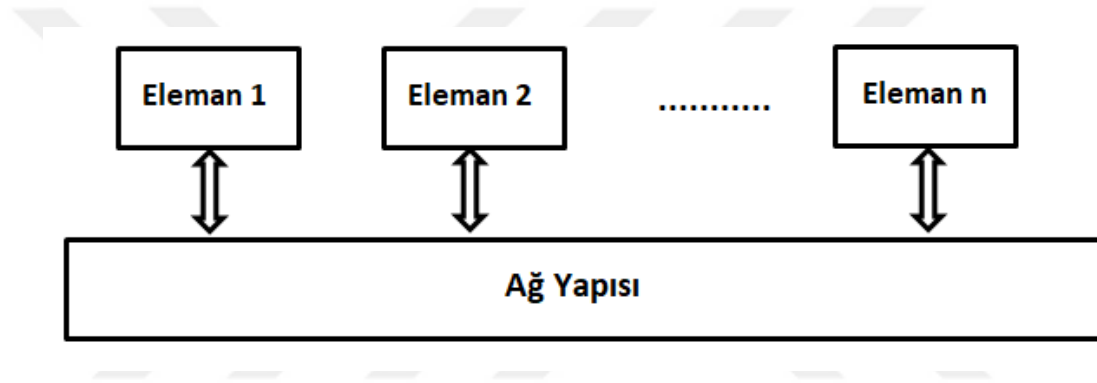
$$\dot{x}_i(t) = - \sum_{j \in N_i(t)} \alpha_{ij}(t) (x_i(t) - x_j(t)) \quad (2.9)$$

Denklemdeki $N_i(t)$ değeri, çok etmenli sistem yapısındaki i elemanı için t anında bilgisine ulaşabildiği diğer elemanların kümesini temsil eder. Bu sistem yapısındaki her bir elemanın durum bilgisi, her seferinde komşularının bilgisine doğru yönlendirilir. Gruptaki bazı elemanların, bazı zaman aralıklarında diğer elemanlar ile herhangi bir bilgi alışverişi yapamayabileceğini ihmal etmememiz gereklidir. [16]'ya göre; iş birliği protokolü $\dot{x} = -Lx$ şeklinde matris formda yazılabilir. Buradaki L , Laplacian matrisini temsil eder. x , $x = [x_1, \dots, x_n]^T$ şeklinde gösterilir.

Ayrık-zamanda iş birliği protokolü ise aşağıdaki şekilde özetlenebilir [16]:

$$x_i[k + 1] = \sum_{j \in N_i[k] \cup \{i\}} \beta_{ij}[k] x_j[k] \quad (2.10)$$

$j \in N_i[k] \cup \{i\}$ için, $\beta_{ij}[k] > 0$ ve $\sum_{j \in N_i[k] \cup \{i\}} \beta_{ij}[k] = 1$ olur. Başka bir deyişle, çok etmenli sistem yapısındaki her bir elemanın bir sonraki durumu, komşularının mevcut durumu ve kendisinin mevcut durumunun ağırlıklı ortalamasına göre güncellenir. Eğer herhangi bir adımda diğer elemanlarla bilgi alışverişi yapmayan bir eleman varsa, bu eleman mevcut durumunu koruyacaktır. Ayrık-zamanda iş birliği protokolü $x[k + 1] = D[k]x[k]$ şeklinde matris formunda yazılır. Buradaki $D[k]$, stokastik bir matristir ve pozitif elemanlara sahiptir. Kısaca iş birliği, $t \rightarrow \infty, \forall i \neq j$ için $\|x_i - x_j\| \rightarrow 0$ olduğu durumda sağlanır [16].



Şekil 2.5 : Çok sayıda elemandan oluşan ağ yapısı [16].

Çok sayıda elemandan oluşan ağ yapısı aşağıdaki özelliklere sahiptir [17]:

- Otonomi: Çok etmenli sistem yapısını oluşturan elemanlar en az yarı-otonom yapıdadır.
- Yerel Görüş: Sistemi oluşturan elemanların hiç biri sistemin bütünü hakkında, yani sistem yapısındaki bütün elemanların bilgilerine sahip olamaz.
- Sorumluluğun Dağıtılması veya Yerel Yönetim Sistemleri: Çok etmenli sistem yapısını oluşturan elemanlardan her biri, kendi durum bilgisiyle göreceli olarak sadece belirli komşu elemanlarla ilişkiye geçer.
- Zamanla Değişim Geçirme: Çok etmenli sistem yapısını oluşturan elemanların her birinin durum bilgisi, belirli iletişim protokollerine bağlı olarak iletişime geçtiği diğer elemanlara göre değişir. Zaman içinde bütün sistem kolektif bir şekilde hareket etmeye başlar.

Çok etmenli sistem yapısındaki elemanların belirli bir değerde iş birliğine varıp ortak hareket etmelerini sağlayan denklem ve algoritmaları daha öncede bahsedilen graf Laplacian terimine göre aşağıdaki şekilde sınıflandırabiliriz [17]:

- Sıradan Laplacian Temelli Protokoller: Sıradan Laplacian, pozitif ve reel değerlere sahip bir graf ile ilişkili bir Laplacian matrisini temsil eder. İş birliği, dizilim kontrolü ve görev dağılımı gibi problemler sıradan Laplacian temelli protokoller yardımıyla çözülebilir.
- İşaret Laplacian Temelli Protokoller: İşaret Laplacian, pozitif veya negatif değerlere sahip bir graf ile ilişkili bir Laplacian matrisini temsil eder. Çift taraflı (bipartite) iş birliği, küme (cluster) iş birliği, dizilim kontrolü ve mesafe temelli konumlandırma problemleri işaret Laplacian temelli protokoller ile çözülebilir.
- Kompleks Laplacian Temelli Protokoller: Kompleks Laplacian, kompleks değerlere sahip bir graf ile ilişkili bir Laplacian matrisini temsil eder. Dizilim kontrolü, göreceli pozisyon temelli konumlandırma ve kompleks nöral ağların çözümünde kompleks Laplacian temelli protokoller kullanılabilir.
- Genelleştirilmiş Laplacian Temelli Protokoller: Genelleştirilmiş Laplacian, matris, zamana-bağlı değişkenler veya dinamik sistemler gibi değerlere sahip bir graf ile ilişkili bir Laplacian matrisini temsil eder. Bu yaklaşım, değişken topolojiye sahip sistemlerde kullanılabilir.

Bir $G = (V, E)$ grafı üzerinde komşularıyla olan yerel iletişimleri sayesinde iş birliğine ulaşmaya çalışan ve $\dot{x}_i = u_i$ dinamiğine sahip elemanlardan oluşan bir çok etmenli sistem yapısı düşünelim. İş birliğine varmayı, aşağıdaki şekilde belirtildiği gibi, bir boyutlu uzayda belirli bir değere yakınsamayı aşağıdaki şekilde açıklayabiliriz [1]:

$$x_1 = x_2 = \dots = x_n \quad (2.11)$$

Bu bir boyutlu uzay $x = \alpha \mathbf{1}$ olarak tanımlanabilir. Bu tanımdaki $\mathbf{1} = (1, \dots, 1)^T$ ve $\alpha \in R$ ise, sistemdeki elemanların ortak kararı olarak belirtilebilir. Sistem yapısında bulunan i elemanı eğer j komşusu ise onunla iletişime geçebilir.

Yönsüz bir grafta (bütün i, j değerleri için $a_{ij} = a_{ji}$), bütün düğümlerin yani sistemdeki elemanların durum bilgisi değişmeyen sabit bir değere eşittir, yani

$\sum_i \dot{x}_i = 0$ olur. Özellikle bu koşulu $t = 0$ ve $t = \infty$ 'da uygularsak, [1]'e göre aşağıdaki sonuca ulaşırız:

$$\alpha = \frac{1}{n} \sum_i x_i(0) \quad (2.12)$$

Başka bir deyişle, eğer iş birliğine asimptotik olarak varılırsa, ağdaki elemanların anlaştığı ortak değer, bütün elemanların başlangıçtaki durum bilgilerinin ortalamaları olacaktır. Bu özellik ortalama-iş birliği algoritması olarak adlandırılır [1].

n tane elemandan oluşan çok etmenli bir sistemde, digraftaki (yönlü graf) her bir düğüm noktası bir eleman olarak kabul edilebilir ve bilgi akışının, digraftaki düğümlerin arasındaki yollarla komşu elemanlar arasında gerçekleştiği görülür. Bu yüzden, ağ yapısındaki sistemlerdeki ilişki topolojisi bir $G = (V, E, A)$ digrafı olarak tanımlanabilir.

Çok etmenli sistem yapısındaki elemanların ikinci dereceden-dinamiğe (second-order dynamics) sahip olduklarına düşünelim. Bu durumda her bir eleman aşağıdaki şekilde modellenir [18]:

$$\dot{x}_i = v_i \quad (2.13)$$

$$m_i \dot{v}_i = F_i, \quad i \in I = \{1, 2, \dots, n\} \quad (2.14)$$

Verilen denklemdeki $x_i \in R$ ve $v_i \in R$ değerleri sırasıyla i elemanının konumu ve hızıdır. $m_i > 0$ değeri i elemanının kütlesi iken, F_i değeri i elemanı üzerindeki toplam kuvvettir. Modeldeki F_i toplam kuvvet değeri iki parçadan oluşur. Buna göre [18]:

$$F_i = -\alpha_i v_i + u_i \quad (2.15)$$

Bu denklemdeki u_i değeri dışarıdan gelen kontrol girdisi iken, $-\alpha_i v_i$ ise sürtünmeden kaynaklı hız sönümlenme terimidir ve $\alpha_i > 0$ koşulunu sağlar. Buna göre ikinci-dereceden dinamik sistem aşağıdaki hale gelir [18]:

$$\dot{x}_i = v_i \quad (2.16)$$

$$m_i \dot{v}_i = -\alpha_i v_i + u_i, \quad i \in I = \{1, 2, \dots, n\} \quad (2.17)$$

Dışarıdan kontrol girdisi olan sistemler için önemli olan gecikme (delay) girdisinin eklenmesiyle birlikte yukarıda belirtilen denklemler aşağıdaki hale gelir:

$$\dot{x}_i(t) = v_i(t) \quad (2.18)$$

$$m_i \dot{v}_i(t) = -\alpha_i v_i(t) + u_i(t - T_i), \quad i \in I = \{1, 2, \dots, n\} \quad (2.19)$$

Denklemden verilen T_i , i elemanı için gecikme(delay) girdisini temsil eder. Çok etmenli bir sistemde bulunan pozisyon durumları temel alınarak oluşturulan iş birliği protokolü aşağıdaki şekilde gösterilebilir [18]:

$$u_i = K_i \sum_{j \in N_i} a_{ij} (x_j - x_i) \quad (2.20)$$

Denklemden; $K_i > 0$, N_i , i elemanının komşularını gösterir ve a_{ij} ise $G = (V, E, A)$ digrafındaki komşuluk matrisinin elemanlarını oluşturur. İletişim gecikmeleri (delays) altında yukarıdaki denklem aşağıdaki hale gelir:

$$u_i(t) = K_i \sum_{j \in N_i} a_{ij} (x_j(t - \tau_{ij}) - x_i(t)), \quad (2.21)$$

Denklemden τ_{ij} değeri, i elemanından j elemanına olan iletişim gecikmesidir. Yukarıda verilen eşitlikleri kapalı-döngü (closed-loop) halinde aşağıdaki şekilde gösterebiliriz [18]:

$$\dot{x}_i(t) = v_i(t) \quad (2.22)$$

$$m_i \dot{v}_i(t) = -\alpha_i v_i(t) + K_i \sum_{j \in N_i} a_{ij} (x_j(t - T_i - \tau_{ij}) - x_i(t - T_i)) \quad (2.23)$$

Lineer elemanlardan oluşan çok etmenli bir sistem yapısında, sürekli-zamandaki iş birliği algoritması aşağıdaki şekilde verilebilir [19]:

$$u_i(t) = \sum_{j \in N_i(t)} a_{ij}(t) (x_i(t) - x_j(t)) \quad (2.25)$$

Denklemden $N_i(t)$ değeri, i elemanı için bilgisine ulaşabildiği elemanların oluşturduğu kümedir. Eğer $\forall i \neq j$ için $t \rightarrow \infty$ iken, $\|x_i(t) - x_j(t)\|$ koşulu sağlanırsa, sistemdeki elemanların iş birliğine ulaşabilecekleri söylenir.

Birinci-dereceden modele ek olarak, ikinci dereceden modeli ele alırsak, aşağıdaki eşitlikleri değerlendirmemiz gerekir [19]:

$$\begin{cases} \dot{x}_i(t) = v_i(t) \\ \dot{v}_i(t) = u_i(t) \end{cases} \quad (2.26)$$

Verilen eşitliklerdeki $\dot{x}_i(t)$, $v_i(t) \in R^n$ değerleri durum vektörlerini ifade eder. $u_i(t) \in R^n$ ise kontrol girdisidir. Bu durumda, lineer elemanlardan oluşan ağ yapısındaki iş birliği algoritması aşağıdaki şekilde olur [19]:

$$u_i(t) = - \sum_{j \in N_i(t)} a_{ij}(t) [(x_i(t) - x_j(t)) + \alpha(t) (v_i(t) - v_j(t))] \quad (2.27)$$

2.4 Çok Etmenli Sistem Uygulamaları

Bu bölümde, çok etmenli bir sistemdeki eleman veya araçların iş birliğiyle ilgili yapılan çalışmaların ve bunun sonucunda elde edilen algoritma ve denklemlerin hangi alanlarda ve nasıl kullanılacağından bahsedilecektir.

Bir sistemin boyutu arttıkça, bu sistem modelleme açısından daha karmaşık hale gelir. Bunun sonucunda da merkezden yani tek bir yerden böyle bir sistemi yönetmek zahmetli olur. Bunun yerine bir ağ yapısına benzeyen çok etmenli sistem yapısı kullanılır. Bu yapıda, yapay zekanın da gelişmesiyle birlikte sistemi oluşturan elemanların her birisinin veya çoğunun kontrol mekanizmasına katıldığı bir mantık izlenir.

Çok sayıda elemandan oluşan sistemlerin uygulamalarına örnek olarak, pazarlama simülasyonları, görüntüleme sistemleri, sistemdeki problemleri tanımlayan ve iyileştiren yapılar verilebilir [20].

Çok etmenli sistem yapısını oluşturan elemanların kontrolü, mobil robotlar, insansız hava araçları, uydular, savaş uçakları, su altı otonom araçları gibi dizilim kontrolüne ihtiyaç duyulan sistemlerde veya görev paylaşımı ve görev atama gibi dizilim kontrolü gerektirmeyen sistemlerde kullanılır [16].

İş birliği algoritmalarının mühendislikteki uygulamalarına [21]'de de değinilmiştir. Buna göre, bir ortamdaki sensor ağlarının kullanımında, otonom olarak yük taşınmasında ve insansız-silahlı araçların kullanılması için askeri alanlarda kullanılır.

Çok sayıda elemandan oluşan sistemlerin iş birliği aynı zamanda endüstri ortamında da çokça kullanım alanı bulmaktadır. Ürün yaşam döngüsü yönetim alanındaki (product lifecycle management-PLM) bir çok parametrenin analiz edilip daha verimli ve etkin üretimin sağlanması bu sayede mümkün olmaktadır [22].

Otonom araç veya elemanlardan oluşan sistemler çok sayıda problem çözmek için geliştirilmiştir. Bunlardan birkaçı; bir ortamın gözetlenmesi, bitkisel ilaçlama, haritalama, hedef belirleme ve izleme, arama ve kurtarma çalışmaları şeklinde sıralanabilir.

Birden fazla elemandan oluşan sistem yapılarını kullanmanın temel amacı; tek bir elemanın hatasının güçlü bir ağ sistemi sayesinde telafi edilmesi ve bu sayede daha zor ve kompleks görevlerin başarıyla yerine getirilmesidir.

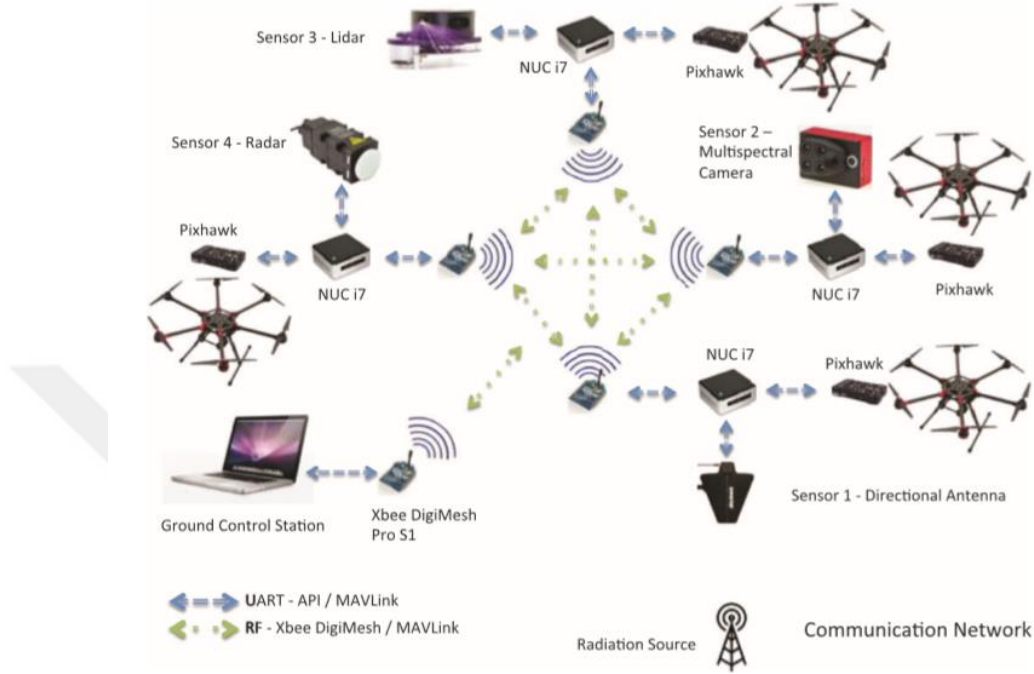
Aşağıdaki kısımda, [23]'de gerçekleştirilen, gerçek dünyaya ait bir senaryo anlatılacaktır. Bu çalışmada değişken (switching) ağ topolojisine sahip bir sistemdeki elemanlara, dağıtılmış (distributed) iş birliği algoritması uygulanması hedeflenmektedir. Senaryoda seçilen uygulamada, canlı bir varlık tespit etme ve izleme görevini yerine getirmek için, insansız ve otonom elemanlardan oluşan bir ağ yapısının kullanılması amaçlanmaktadır. İlgili senaryo, robotlardan oluşan bir ekibin, doğal bir felaketten sonra kayıp bir kişiyi bulmak için görevlendirilmesi olarak düşünülebilir.

Gerçek zamanlı uygulamanın karmaşıklığını azaltmak için, çok sayıda elemandan oluşan ağ yapısının canlı bir varlık arama-tespit etme ve takip görevlerini yerine getirme görevi beş parçaya bölünmüştür. İzlenecek yol beş aşamada anlatılacaktır.

Birinci aşamada iletişim ele alınır. Buna göre ağ yapısındaki elemanlar için bir iletişim protokolü geliştirilir. Yani ele alınan ilk görev, komşu elemanlar arasında bir iletişim protokolünü oluşturulmasıdır. Şekil 2.6, ağdaki her bir elemanın ihtiyaç duyduğu bilgilerin iletilmesini ve alınmasını sağlamak için uygulanan iletişim şemasını göstermektedir [23]. Ağdaki kafes yapısının temel amacı, ağdaki elemanlar arasında bilgiyi paylaşmaktır.

İkinci aşamada tahmin geliştirilip, canlı bir varlığın hangi yönde olduğu belirlenir. Çok etmenli bu ağ yapısındaki her bir eleman, canlı varlığın bulunduğu yöne bağlı bir yörünge açısı (bearing angle) tahmin etmek için kendi üzerindeki sensörlerden faydalanır. Bu aşamada algılayıcının (sensing device), kaynağın geldiği yöne karşılık

gelen bir yörünge açısı (bearing angle) belirleyebilmesi için, sistemdeki elemanın kendi eksenini (dikey eksen) etrafında bir kaç kez dönebilmesini sağlayan bir kontrol stratejisi uygulanır. Şekil 2.7, sistemdeki elemanların bu aşamadaki davranışlarını göstermektedir [23].

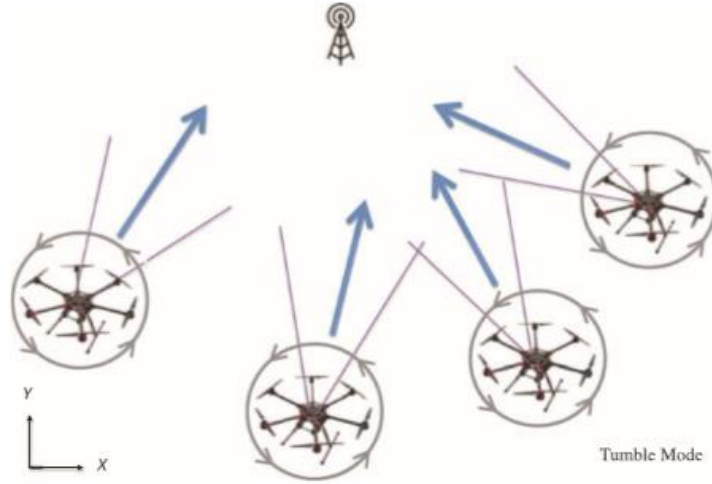


Şekil 2.6 : Her bir elemanın ihtiyaç duyduğu bilgi alışverişini sağlayan iletişim ağı gösterilmektedir [23].

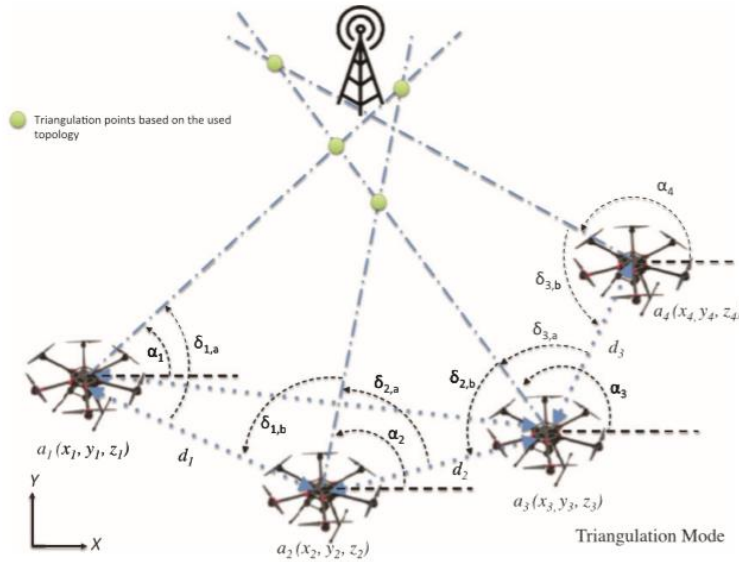
Üçüncü aşamada canlı varlığın bulunduğu koordinatlar tahmin edilir. Söz konusu çok etmenli sistemde her bir eleman canlı varlığın bulunabileceği yöne bağlı olarak yörünge açısını tanımlayan bir tahminde bulunduğu anda, kendi koordinatlarıyla birlikte bu bilgiyi de komşularına iletir. Bu, tüm mevcut ölçümleri bir araya getirerek, düzlemsel koordinatlarda canlı varlığın konumunun doğru tahmin edilmesini sağlar. Sistemdeki diğer komşu elemanların ilgili yörünge açıları tahminlerini bir araya getiren algoritma, genişletilmiş Kalman filtresinin (Extended Kalman Filter) bir uygulaması olarak gerçekleştirilir. Şekil 2.8, bu durumu anlatmaktadır [23].

Dördüncü aşamada canlı bir varlığa ulaşmak için geliştirilen iş birliği algoritması uygulanır. Bu bölümde, canlı varlığın (X, Y) düzlemindeki koordinatlarından faydalanılarak, iş birliği algoritması yardımıyla her bir elemanın pozisyonunu canlı varlığın etrafında olacak şekilde stabilize etmesi hedeflenir. Canlı varlığın koordinatları, sistemdeki lider elemanın koordinatları olarak kullanılır. Uygulanan

kontrol ile birlikte, ağ yapısındaki elemanların canlı varlığa doğru hareket edip ona ulaşmaları ve hatta gerekiyorsa ona eşlik etmeleri sağlanacaktır. Şekil 2.9, ağdaki elemanların bir takım halinde organize olarak canlı varlığa ulaşmalarında kullandıkları özel yörüngeleri göstermektedir [23].



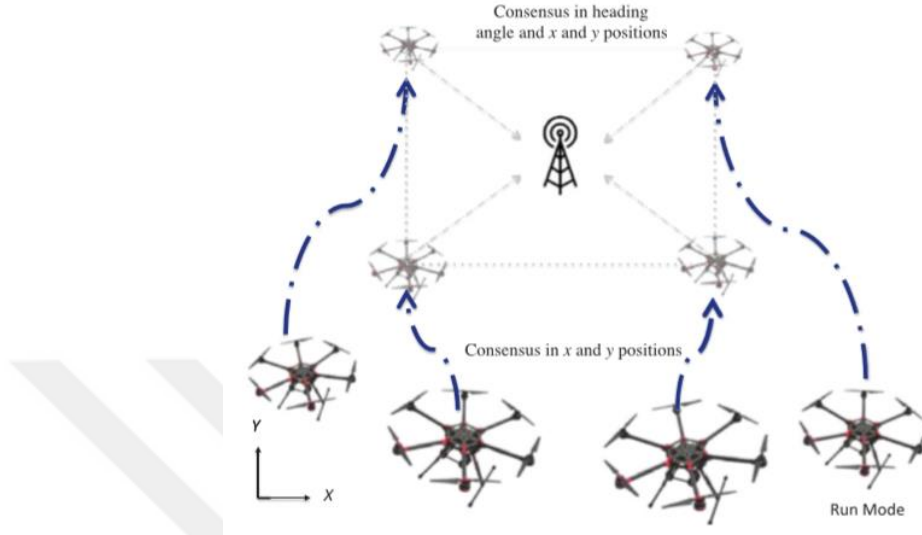
Şekil 2.7 : Ağ yapısındaki her bir eleman kendi ekseninde dönerek canlı varlığın bulunduğu doğru açıyı bulmaya çalışmaktadır [23].



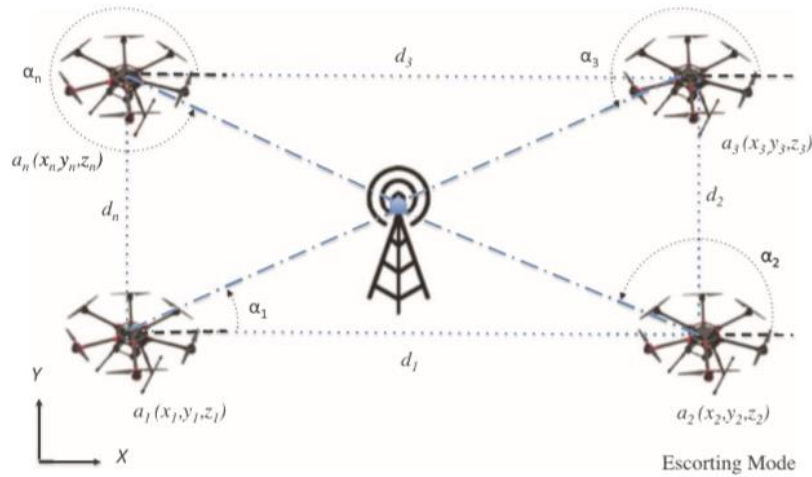
Şekil 2.8 : Ağ yapısındaki elemanlar, canlı varlıktan gelen sinyal ile komşularından aldığı pozisyon bilgilerini bir araya getirerek belirli bir sürüş açısı elde etmeye çalışmaktadır [23].

Beşinci ve son aşamada, canlı varlığın izini sürme, gözlem yapma ve takip etme süreçleri gerçekleştirilir. Bu bölümdeki işlemlerde gerçek-zamanlı (real-time)

uygulamaların gerçekleştirilebilmesi için kritik olan değişken (switching) iş birliği algoritması temel alınır. Canlı varlığa eşlik etme görevi yerine getirilirken, sistemdeki her bir eleman, (X,Y) düzleminde belirli bir koordinat ve açı belirleyerek canlı varlıkla arasında belirli bir mesafe kalacak şekilde hareketini gerçekleştirilir.



Şekil 2.9 : Her bir eleman komşularından aldığı bilgiyi de kullanarak belirli bir koordinat doğrultusunda canlı varlığa doğru gitmektedir [23].



Şekil 2.10 : Ağ yapısındaki elemanlar verilen görevi takım halinde çalışarak iş birliği çerçevesinde yerine getirmektedir [23].

Önerilen yaklaşım, olası bir iletişim kaybı senaryosunda, canlı varlığın tespitinde kullanılan sensörlerin bozulmasında veya sistemdeki elemanların görüş alanında bir engel olması durumunda bile verilen görevin düzgün bir biçimde gerçekleştirilmesini

sağlayacaktır. Şekil 2.10, canlı varlığa ulaşan elemanların uçuş dizilimini ve bu elemanların eşlik etme görevlerini nasıl yerine getirdiklerini göstermektedir [23].



3. KALMAN FİLTRESİ

3.1 Olasılığa Giriş ve Normal Dağılımlar

İlerleyen bölümlerde, olasılık ve rastgele değişkenler (random variables) hakkında bilgiler verilecektir.

Olasılık genel anlamda, belirli bir olayın gerçekleşme ihtimali olarak tanımlanabilir. Buna göre, belirli bir A olayının gerçekleşme olasılığı aşağıdaki gibidir [24]:

$$p(A) = \frac{\text{A'nın lehine olan muhtemel sonuçların sayısı}}{\text{muhtemel sonuçların toplam sayısı}} \quad (3.1)$$

Bir sonucun A veya B olayının lehine olma olasılığı aşağıda verilmiştir [24]:

$$p(A \cup B) = p(A) + p(B) \quad (3.2)$$

Eğer iki olayın olması olasılığı birbirinden bağımsız ise bu iki olayın gerçekleşme olasılığı, tek başına sahip oldukları olasılıkların çarpımına eşittir [24]:

$$p(A \cap B) = p(A)p(B) \quad (3.3)$$

B olayının gerçekleştiği durumda, A olayının gerçekleşme ihtimali ise, koşullu olasılık olarak adlandırılır [24].

$$p(A|B) = \frac{p(A \cap B)}{p(B)} \quad (3.4)$$

Rastgele değişken (random variable) genel olarak bir örnek uzaydaki bütün noktaları reel sayılara dönüştüren bir fonksiyondur. Örneğin, sürekli-rastgele değişken (continious random variable) $X(t)$ zamanı pozisyon bilgisine dönüştürebilir. Zamandaki herhangi bir noktada, $X(t)$ beklenen pozisyon hakkında bilgi verebilir.

Sürekli-rastgele değişkenler açısından, herhangi bir A ayrık olayının olasılığı sıfırdır ve bu durum $p(A) = 0$ şeklinde gösterilir [24]. Bunun yerine olasılıkları belirli bir aralıkta değerlendirme yoluna gidilir. Rastgele değişkenlerin olasılığını gösteren

temel fonksiyon, kümülatif dağılım fonksiyonu (cumulative distribution function) olarak adlandırılır ve aşağıdaki şekilde gösterilir [24]:

$$F_X(x) = p(-\infty, x] \quad (3.5)$$

Bu fonksiyon x de dahil olmak üzere x 'e kadar olan bütün olaylar için sürekli-rastgele değişken X için kümülatif olasılık değerini belirtir. Kümülatif dağılım fonksiyonunun bazı önemli özellikleri aşağıda verilmiştir [24]:

- $x \rightarrow -\infty$ iken $F_X(x) \rightarrow 0$ olur.
- $x \rightarrow +\infty$ iken $F_X(x) \rightarrow 1$ olur.
- $F_X(x)$, x 'in azalmayan bir fonksiyonudur.

Kümülatif dağılım fonksiyonunun daha da fazla yaygın olan hali, bu fonksiyonun türevidir ve bu da olasılık dağılım fonksiyonu (probability density function) olarak bilinir [24].

$$f_X(x) = \frac{d}{dx} F_X(x) \quad (3.6)$$

Olasılık dağılım fonksiyonu aşağıdaki özelliklere sahiptir:

- $f_X(x)$ negatif olmayan bir fonsiyondur.
- $\int_{-\infty}^{\infty} f_X(x) dx = 1$

Sonuç olarak, belirli bir $[a, b]$ aralıktaki olasılık aşağıdaki gibi tanımlanır [24]:

$$p_X[a, b] = \int_a^b f_X(x) dx \quad (3.7)$$

Ayrık olay denklemlerindeki gibi olasılıkların toplanması yerine, sürekli-rastgele değişkenler için olasılık dağılım fonksiyonu kullanılarak ilgili aralıkta integral alınır [24].

X bir küme ve F de X 'in alt kümelerinin bir toplamı olsun. (X, F) üzerindeki bir olasılığın ölçüsü, $\mu: F \rightarrow [0,1]$ şeklinde bir fonksiyondur [25]. Başka bir deyişle, μ , F 'deki her küme için, 0 ile 1 arasında bir olasılık değeri belirler. μ 'nin etki alanı bütün kümelerin bir toplamı olduğu için μ 'ya küme fonksiyonu denir. μ 'nın bir olasılık belirtebilmesi için ise aşağıdaki koşulları sağlaması gerekir [25]:

- $\mu(\emptyset) = 0$ ($\emptyset$, boş kümedir)
- $\mu(X) = 1$, ve
- Eğer A_1 ve A_2 ayrık ise $\mu(A_1 \cup A_2) = \mu(A_1) + \mu(A_2)$

X , sonlu veya sayılabilir sonsuz bir küme olduğunda, μ 'ya kesikli olasılık (discrete probability) denir. X , bir aralık değeri aldığı anda ise, sonlu veya sonsuz olduğuna bakılmaksızın, μ 'ya sürekli olasılık (continuous probability) denir. Kesikli durumda, F , X 'in olası bütün alt kümelerini içerir. Fakat sürekli durumda, F 'in X 'in olası bütün alt kümelerini içermesi durumu gerçekleşmez.

Pratikte, X , bir denemenin olası bütün sonuçlarının kümesidir. μ değeri ise deneme sonucu gerçekleşmesi muhtemel bir olayın örneklemesidir. Örneğin bir zar atıldığında, sonuçlar kümesi $\{1,2,3,4,5,6\}$ olduğundan $X \equiv \{1,2,3,4,5,6\}$ olarak belirlenir. Eğer standart ve düzgün bir zar olduğuna inanıyorsak, $\mu(\{1\}) = \mu(\{2\}) = \dots = \mu(\{6\})$ şeklinde gerçekleşir. Diğer olası değerleri uygulayacak olursak, aşağıdaki değerleri elde ederiz:

$$\mu(\{1,2\}) = 1/3$$

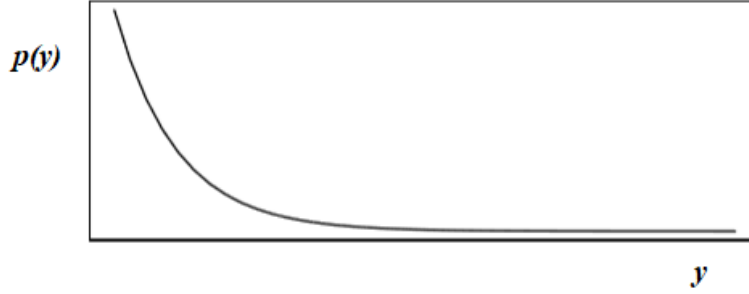
$$\mu(\{2,4,6\}) = 1/2$$

Genelde μ terimi yerine P sembolü veya olasılık kelimesi kullanılmaktadır. Olasılık ölçüsü terimi yerine ise dağılım (distribution) tercih edilmektedir.

Şu ana kadar kesikli-olasılıklarla, yani sayılamayan sonuçlar için olasılıkları ele aldık. Aşağıda örnekleri verilen durumlarda ise X belirli bir aralıkta yer almaktadır [25].

- Tıbbi tedavi: Bir hasta için hastalığının tekrar ortaya çıkmasına kadar geçen süre
- Spor: Cirit atma uzunluğu
- Ekoloji: Bir ağacın yaşam döngüsü
- İmalat: Bir rulmanın çap uzunluğu
- Bilgi işlem: Müşteri temsilcisiyle görüşen bir müşterinin hatta beklediği süre
- Fizik: Bir Uranyum atomu için bozulana kadar geçen süre
- Okyanus bilimi: Belirli bir enlem, boylam ve derinlikteki okyanus suyunun sıcaklığı

Yukarıda örneklerdeki değişkenlerin çıktıları sürekli (continious) olarak adlandırılır. Örneğin, Y , müşteri temsilcisiyle yapılan görüşmede hatta beklenen süre olsun. Random değişken (random variable) Y genellikle Şekil 3.1'e benzer bir yoğunluk (density) ile modellenir [25].



Şekil 3.1 : Hatta beklenen süre için PDF(probability density function) [25].

Şekil 3.1'de gösterilen eğri olasılık dağılım fonksiyonunu (probability density function) gösterir. Bu fonksiyonu göstermek için genelde, p, π veya f kullanılır. Bir olasılık dağılım fonksiyonu için olasılığı eğrinin altında kalan alan belirtir. Örneğin, müşteri temsilcisini bekleme süresi 60 dakikadan az olan müşterilerin olasılığı aşağıdaki şekilde gösterilir [25]:

$$P[Y < 60] = \int_0^{60} p(t)dt$$

Bir olasılık dağılım fonksiyonu aşağıdaki iki özelliği sağlamalıdır [25]:

- Bütün y 'ler için $p(y) \geq 0$
- $\int_{-\infty}^{\infty} p(y)dy = 1$

PDF'ler olasılıkların türevleridir. Herhangi bir a sabit sayısı için [25]:

$$\frac{d}{db} P[X \in (a, b)] = \frac{d}{db} \int_a^b f_X(x)dx = f_X(b) \quad (3.8)$$

Benzer şekilde, $d/da P[X \in (a, b)] = -f_X(a)$ olur.

Olası bütün olasılık ölçümlerinin dağılım (distribution) kümesi olasılık ölçümlerinin bir parametrik ailesi (parametric family) olarak adlandırılır. Aşağıdaki paragraflarda bu olasılık ölçümlerinin en çok kullanılan parametrik aileleri (dağılımlar) hakkında bilgiler verilecektir.

Binom dağılımında (binomial distribution), istatistikçiler aşağıdaki tipteki gözlemleri dikkate almak durumundadır [25]:

- Tekrar edebilen bir olay ya başarı ya da başarısızlıkla sonuçlanır.
- Çok sayıda tekrar edilerek gözlem yapılır.
- Başarı ve başarısızlık sayıları tutulur.
- Başarıların sayısı, başarı olasılığı hakkında bilgi edinmemize yardımcı olur.

Bu tarz gözlemler, binom (binomial) olarak adlandırılır. Bazı kullanım alanları aşağıda verilmiştir:

- Tıbbi alandaki tedavi denemeleri: Aynı tedavi çeşidi veya aynı ilaç birden çok hastaya uygulanır ve bazıları bu sayede iyileşirken bazıları iyileşmez.
- Toksisite testleri: Birçok laboratuvar hayvanı kanserojen maddeye maruz kalır. Bazıları kanser olurken bazıları olmaz.

Binom dağılımında olasılıklar aşağıdaki denkleme göre hesaplanır.

$$P[X = k|\theta] = \binom{N}{k} \theta^k (1 - \theta)^{N-k} \quad (3.9)$$

İlgili denklemde binom katsayısı $\binom{N}{k}$, deneme sayısı N , başarı olasılığı p veya θ , k 1'den N 'e kadar olan bir değeri belirtir ve başarılı olma sayısı ise X ile gösterilir.

Poisson dağılımında, aşağıda belirtilen tipte gözlemler yapılır [25]:

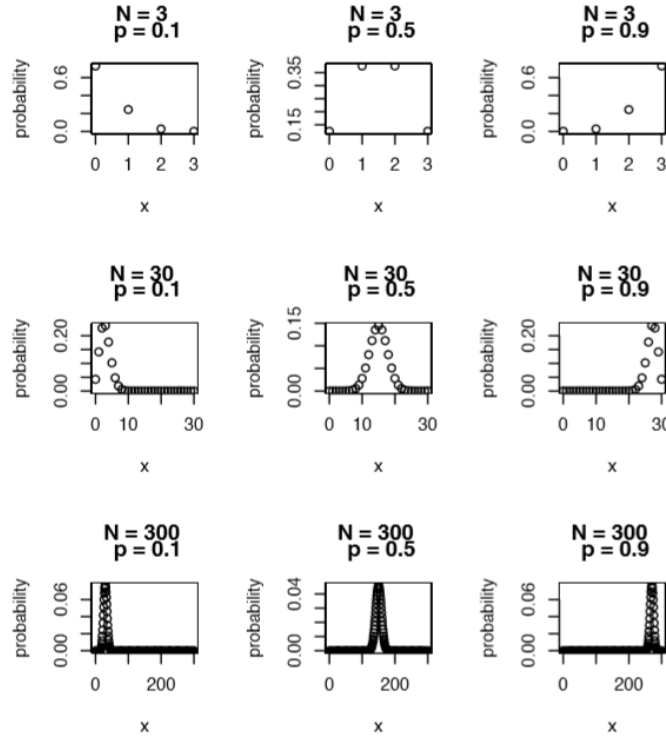
- Genellikle belirli bir çalışma alanı vardır.
- İlgili alandaki olaylar rastgele olacak şekilde gerçekleşir.
- Olayların gerçekleşmesi temel bir orana göre olur.

Bu tarz gözlemler Poisson olarak adlandırılır. Çalışma alanındaki olayların sayısı ilgili oran hakkında bilgi edinmemize yardımcı olur. Bazı örnekler aşağıdaki gibidir:

- Ekoloji: Ağaç fidelerinin ormanda yetişmesi oranı
- Bilgisayar programlama: Bilgisayar kodunda hatalar oluşması oranı
- Genetik: Gen diziliminde mutasyonların gerçekleşmesi oranı

Poisson dağılımında olasılıklar aşağıdaki denkleme göre hesaplanır [25]:

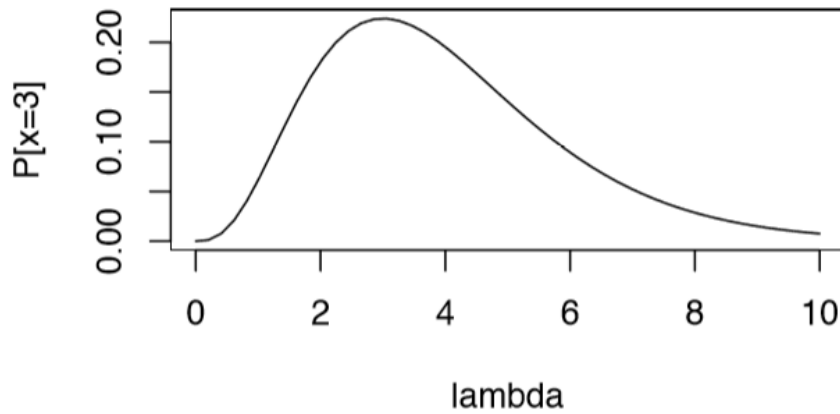
$$P[X = k|\lambda] = \frac{\lambda^k e^{-\lambda}}{k!} \quad (3.10)$$



Şekil 3.2 : Binom dağılımındaki olasılıkların gösterimi [25].

İlgili denklemde, λ bir olayın gerçekleşme oranını ve X ise olayın gerçekleşme sayısını gösterir.

$P[X = 3|\lambda]$ olan bir Poisson dağılımı aşağıdaki şekilde gösterilmiştir [25].



Şekil 3.3 : Poisson dağılımında olasılıkların gösterimi [25].

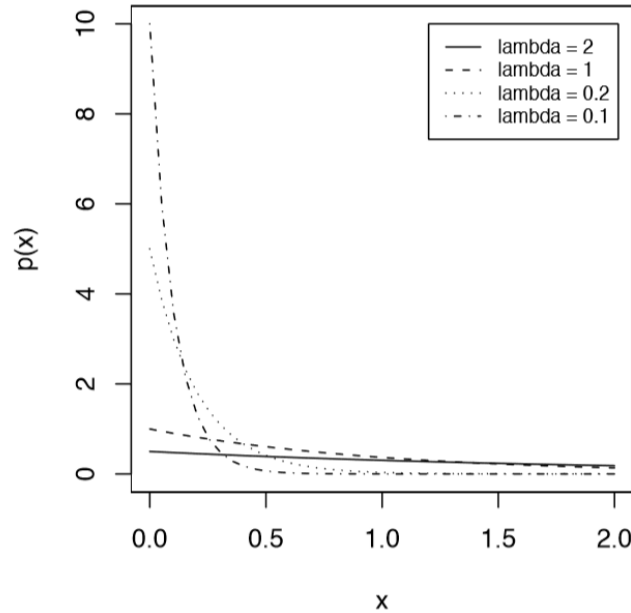
Üstel (exponential) dağılımda ise, yoğunluğu (density) sıfırdan giderek uzaklaşan bir sürekli-rastgele (random) X değişkeni için modelleme yapılır [25]. Aşağıda kullanım alanlarına örnekler verilmiştir:

- Müşteri hizmetleri: Bir müşterinin, müşteri hizmetlerine bağlanmak için hatta beklediği süre
- Nörobiyoloji: Bir sonraki nöronun ölmesine kadar geçen süre
- Sismoloji: Bir sonraki depreme kadar geçen süre
- Tıp: Bir kanser hastası için kalan ömür

Bu gibi durumlar için kullanılan olasılık dağılım fonksiyonu üstel yoğunluk olarak adlandırılır ve tanımı aşağıda verilmiştir [25]:

$$x > 0, \quad p(x) = \frac{1}{\lambda} e^{-\frac{x}{\lambda}} \quad (3.11)$$

Denklemden, X λ parametresine bağlı bir üstel dağılıma sahiptir. Aşağıdaki şekilde çeşitli λ değerlerine bağlı üstel yoğunluklar gösterilmiştir [25].



Şekil 3.4 : Üstel yoğunlukların gösterimi [25].

Normal dağılımda, çan eğrisi benzeri yapıda yoğunluğa sahip bir sürekli-rastgele (random) değişken Y için modelleme yapılır. Kullanım alanlarına örnekler aşağıda verilmiştir [25]:

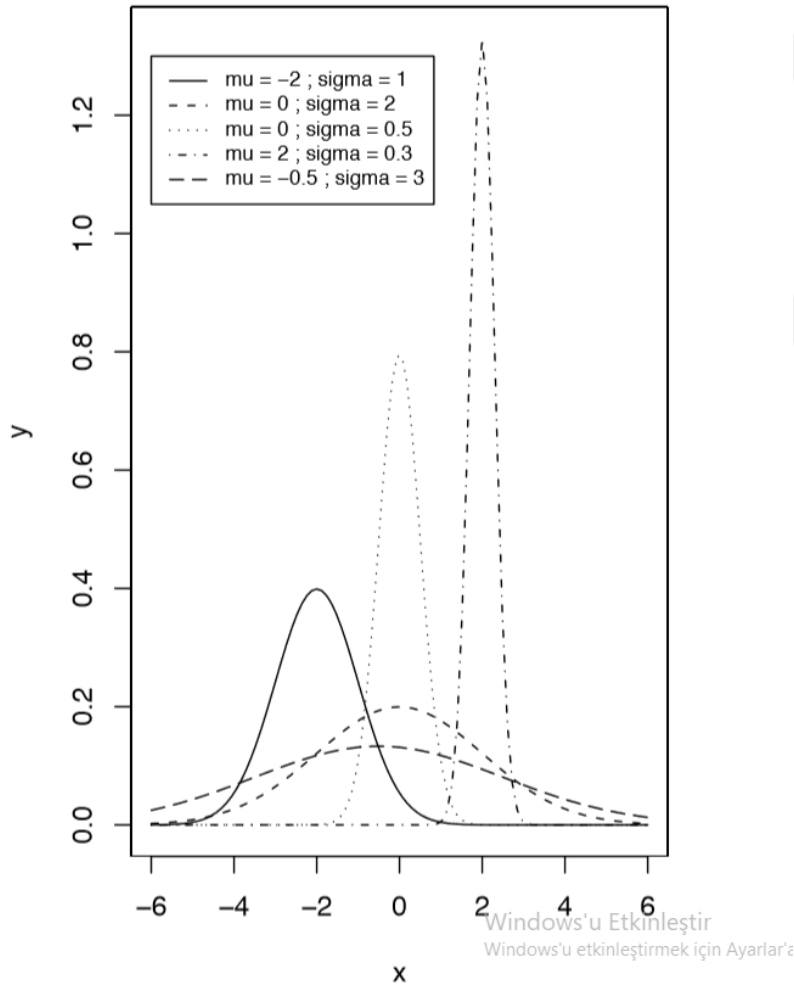
- Biyolojik antropoloji: İnsanların boy uzunlukları

- Okyanus bilimi: Belirli bir alandaki okyanus suyu sıcaklıkları
- Kalite kontrol: Bir rulmanın ap lüsü
- Eēitim: Bir okuldaki ērencilerin sınav sonuçları

Normal yoēunluēun olduēu durumlarda olasılık daēılımı aēaēıdaki gibi olur:

$$p(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2} \quad (3.12)$$

Verilen denklemde, μ ortalama, σ ise standart sapma, olarak bilinir. Aēaēıdaki Őekil normal yoēunluēa rnek olarak gsterilebilir [25].



Őekil 3.5 : Belirli parametrelere gre normal yoēunluēun gsterimi.

Normal daēılımları ilerleyen paragraflarda biraz daha detaylı inceleyeceēiz. Normal eēri, Őekil 3.6'daki gibi an Őeklinde bir eēridir. ok sayıda lüm alındıēı

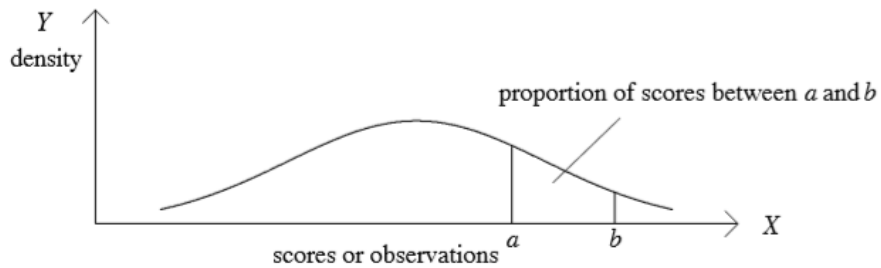
durumlarda bir çok niteliğin yaklaşık olarak dağıtılması nedeniyle, istatistik alanında çok kullanışlı bir eğridir.



Şekil 3.6 : Normal eğri [26].

Normal dağılım modeli, 1733'te, kumar oyunlarının olasılıklarıyla ilgilenen matematikçi Abraham Deoivre'nin çalışmasında ortaya çıkmış ve bundan bağımsız olarak 1786'da bir astronom ve matematikçi olan Pierre Laplace tarafından türetilmiştir [26]. Fakat, normal eğri, daha çok bilimsel teorilerdeki hata dağılımı açısından, normal eğri için yeni bir formül elde eden astronom ve matematikçi olan Karl Friedrich Gauss ile ilişkilidir. Bu nedenle, normal eğri bazen Gaussian eğrisi olarak da bilir. 1835'te bir başka matematikçi ve astronom olan Lambert Qutelet, insan fizyolojisi ve sosyal özelliklerini tanımlamak için bu modeli kullanmıştır.

Bir değişken için normal dağılım çizdiğimizde, değişkenin değerleri X eksenini denilen yatay eksenle temsil edilir. Eğri altında kalan herhangi bir aralıktaki alan, o aralıktaki oranı temsil eder. Bir a ile b aralığı arasındaki eğrinin yükseliği, o aralığın yoğunluğu veya kalabalıklığı olarak adlandırılır ve bir aralıktaki eğri ne kadar yüksekse o aralıktaki yoğunluk o kadar fazla olur [26]. İlgili durum Şekil 3.7'de gösterilmiştir.



Şekil 3.7 : X değişkeninin iki değer aralığı arasındaki puan veya gözlem oranlarının gösterimi [26].

Normal dağılım özelliklere sahiptir [26]:

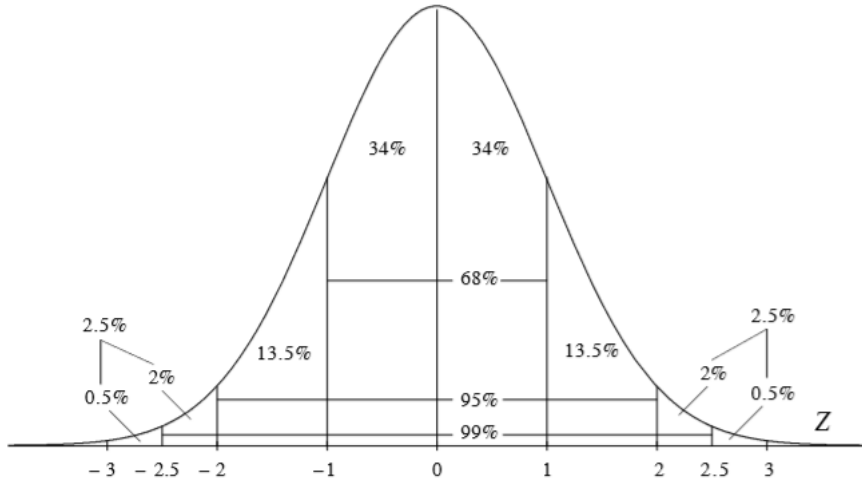
- Eğrinin altında kalan alan 1'dir.
- Eğri simetriktir. Bu yüzden mod, medyan ve ortalama (mean) birbiri üzerindedir.

- Eğri çan şeklindedir.
- En büyük oran, ortalama (mean) değerine çok yakındır.
- Mean değerinden her iki yöne gidildikçe oran azalır.
- Hemen hemen bütün gözlenen değerler (0.997 tanesi) mean değerinden 3 standart sapma değeri kadar uzaklıktaki alanda bulunur.

Normal dağılım gösterimlerine göre, X değişkeni, x ise değişkenin değerini ifade etmektedir. X değişkenine göre normal dağılımda, μ_X ortalama değerini, σ_X ise standart sapma değerini ifade eder. En genel haliyle, $\sim N(\mu_X, \sigma_X^2)$ ile ifade edilir. σ_X^2 ifadesi varyans olarak adlandırılır ve standart sapmanın karesine eşittir.

Standart normal dağılım için, ortalama değeri 0, standart sapma ve varyans değeri ise 1'dir. Eğer Z bir standart normal değişken ise $\mu_Z = 0$, $\sigma_Z = 1$ ve $\sigma_Z^2=1$ olur.

Aşağıdaki şekilde standart sapma değerine göre (yatay eksen) standart normal dağılımın altında kalan alan yaklaşık olarak yüzde biçiminde verilmiştir [26].



Şekil 3.8 : Standart normal eğri altında kalan alanın yüzdesel olarak gösterimi [26].

3.2 Lineer Sistemlerde Tahmin Yöntemlerinin Temelleri

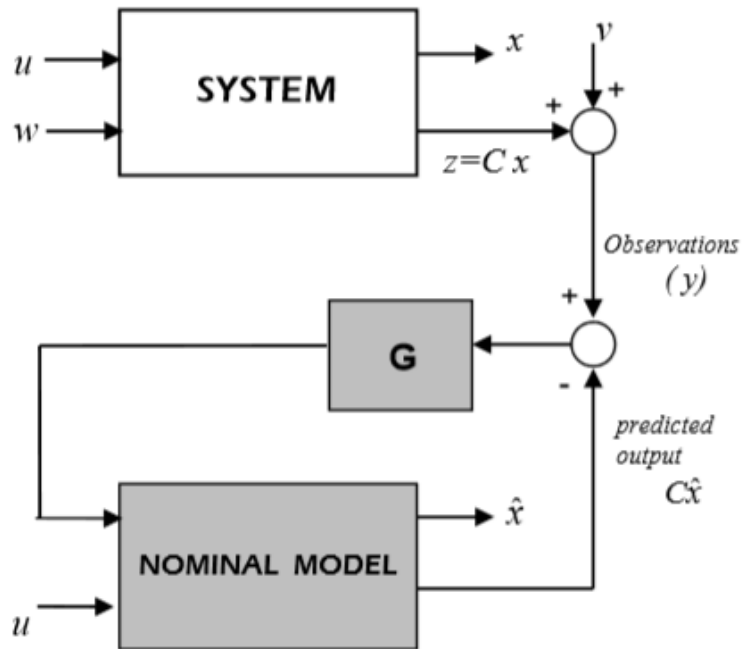
Bir sistemdeki elemanların durum bilgisinin tahmini, bir model ve ölçümlerden oluşan söz konusu sistem hakkında önceden bilinen bilgilerle, tüm durum vektörünün gelişimini tahmin etmek için birleştirildiği bir prosesten oluşur. Tahmin algoritmasını

geliştirmek için kullanılan matematiksel alt yapı gözlemci olarak bilinir ve bazen de bir filtre olarak adlandırılır.

Bir sistemdeki kazanç olarak adlandırılan terimin doğru olması, ölçülemeyen bozucu girdilerin ve modelleme hatalarının tahmin sürecine dahil edilmesiyle sağlanır. Bu durum literatürde proses gürültüsü olarak adlandırılır. Kalman filtresi ve Luenberger gözlemcileri gibi klasik tahmin yöntemlerinde, sadece ölçülemeyen dış bozucular göz önüne alınır ve dolayısıyla sistemin modellemesinin düzgün yapıldığı varsayılır [27].

Şekil 3.9’da bir gözlemci için blok diyagram gösterilmiştir [27]. Buna göre bu şekildeki sistem düşünecek olursa; u belirli sistem girdisi (deterministic input), w ise bilinmeyen bir girdidir (unknown input). Çıkış sinyallerinin z ile gösterildiğini varsayalım. Ölçüm cihazlarının kusurlu olmasından ve bilginin toplanmasında oluşabilecek potansiyel problemlerden dolayı gözlenen değerlerden oluşan z vektörünün ölçüm gürültüsü tarafından bozulduğu görülür. Yani $y = Cx + v$ değeri elde edilir. Diyagramdaki $C\hat{x}$ öngörülen çıkış sinyalini ifade eder ve gözlenen değerden çıkarılarak $(y - C\hat{x})$ kazanç olarak nominal modele girer. Bu sayede $\hat{x}$, tahmin değeri, iteratif olarak kendini yeniler.

Durum bilgisi tahmini yapabilmek için ana amaç, bozulmuş ölçümleri ve nominal modeli en iyi şekilde kullanmaktır.



Şekil 3.9 : Bir gözlemci için blok diyagramın gösterimi [27].

Durum bilgisini tahminini yapabilmek için, her k adımı için bir $x(k)$ durumunun bir $\hat{x}(k)$ tahmini oluşturulur. Bu tahminin oluşturulması için $y(k)$ çıkış değeri ve $x(k)$ girdi değeri kullanılır.

Çok çeşitli tahmin veya filtre yöntemleri arasında, Kalman filtresi bir dinamiğe sahip sistemlerde en çok kullanılan algoritmalarından birisidir. Kalman filtresinin içerisine iş birliği algoritmasının kolaylıkla eklenerek, sistemdeki elemanların bozucu girdilere rağmen iş birliğine varmalarının sağlanması mümkündür [28].

İlerleyen bölümlerde Kalman filtresi ve genişletilmiş Kalman filtresi hakkında detaylı bilgiler verilecektir.

3.3 Kalman Filtresi Algoritması

Bir sistemi analiz ederken veya bir kontrolör tasarlarken, bir mühendisin elinde kontrol teorilerinden ve deterministik (rastlantısız olmayan) sistemlerden türetilen geniş bir bilgi birikimi vardır. Bu durumda doğal olarak, “neden bunların yerine stokastik (rastlantısız) sistem modelleri ve bunların üzerine kurulu tahmine dayalı kontrol yöntemlerine ihtiyaç duyarız?” sorusu sorulabilir. Bu soruyu cevaplamak için, deterministik yöntemlerin neyi yapabildiği ve neyi yapamadığı konuları üzerinde durulması gerekir [29].

Verilen bir fiziksel sistemde, bu fiziksel sistem, bir uçak, bir kimyasal proses veya bir ekonomi sistemi olabilir, bir mühendis ilk olarak bir matematiksel model oluşturarak, bu sistemin davranışıyla ilgili her yönden bilgi sahibi olmak ister. Kontrol teorileri ve sistemleri tarafından sağlanabilen bu matematiksel model ve araçlar, sistemin yapısı ve davranış modları hakkında bilgi sahibi olabilmemizi sağlar [29].

Gerçek sistem cevabını gözlemleyebilmek için, ölçüm cihazları, belirli değişkenlerle orantılı olan çıkış sinyalleri alabilmemizi sağlar. Bu çıkış sinyalleri ve sisteme uygulanan girdiler (inputs) sistem davranışı hakkında direkt olarak farkedilebilen tek bilgilerdir. Üstelik, eğer bir geri bildirim mekanizması tasarlandıysa, ölçüm cihazının çıkışları, kontrolöre giren ve direkt olarak ulaşılabilir olan tek sinyallerdir. Deterministik sistemlerin ve kontrol teorilerinin, bir sistem modelinin tasarımında ve sistemin bu modelle analiz edilmesinde tamamen yeterli olmamasının üç nedeni aşağıda sıralanmaktadır [29].

İlk olarak, hiç bir matematiksel model mükemmel veya kusuruz değildir. Matematiksel model yalnızca mühendislerin ihtiyacına yönelik olacak biçimde karakteristik özellikleri tasvir etmeye yarar. Matematiksel model, sistem için kritik olan belirli koşullar için sistemin gösterdiği cevaplara yöneliktir. Fakat kritik olmayan diğer koşullar modellemenin dışında kalır ve sistemin bu noktalarda davranışı bilinemez.

Deterministik modelin ikinci eksik kaldığı nokta ise, dinamik sistemlerin sadece kendi kontrol girdileri tarafından değil de, aynı zamanda kontrol edilemeyen veya modellenemeyen bozucu girdiler tarafından da etkilenmesidir. Örneğin, bir pilot kullandığı uçağı belirli bir açısal yörüngede kumanda etmeye çalışsa da, rüzgar yönü, motorların hassasiyeti gibi bozucular yüzünden elde edilecek konum daha farklı olacaktır.

Deterministik yöntemlerin ve matematiksel modellemenin eksik kaldığı noktaların üçüncüsü ise, sensör gibi algılayıcıların herhangi bir sistem hakkında tam ve eksiksiz veriye ulaşamamasıdır. Sensörler, bilmek istediğimiz bilgilere tam olarak ulaşmamızı sağlayamamaktadır. Çünkü kendi sistemleri de aynı zamanda belirli bir gürültüye maruz kalmaktadır.

1960 yılında, R.E Kalman kendi makalesinde, lineer filtreleme ve tahmin problemi için, kendini iteratif bir şekilde güncelleyen bir çözüm bulmuştur. Bu andan itibaren, dijital hesaplama yöntemlerinin gelişmesiyle beraber, bir çok uygulama ve araştırma alanında Kalman filtresi yaygın bir biçimde kullanılmaya başlanmıştır.

Kalman filtresi, bir modelin önceki durum bilgisinden yola çıkarak giriş ve çıkış değerleriyle birlikte sistemin durum bilgisini tahmin edebilen bir filtredir. Kalman filtresi algoritması ise gürültülü olan veriler üzerinde iteratif bir şekilde gerçek zamanlı çalışarak hataları minimize eder ve sistemin fiziksel ve matematiksel modeline göre gelecek durumu hakkında tahminler üretir.

Kalman filtresi algoritmasında, gözlenen değer ile tahmin karşılaştırılır ve aradaki fark, Kalman kazancı (Kalman gain) olarak adlandırılır ve bir çarpan ile ölçeklendirilir. Daha sonra bu, sıradaki tahminleri iteratif bir şekilde güncellemek için modele bir geri besleme şeklinde uygulanır. Kalman kazancı ayarlanabilir ve eğer katsayısı yüksek tutulursa, gözlenen duruma daha yakın sonuçlar elde edilir ve eğer katsayısı düşükse de tahmin değerlerine daha yakın sonuçlar elde edilir [30].

Kalman filtresi, durum bilgisi $x \in R^n$ olan ayrık-zamanlı (discrete-time) bir şekilde kontrol edilen bir prosesin tahmin edilmesini amaçlayan genel bir problemin çözümünde kullanılır. Ayrık-zamanlı lineer bir sistemin durum ve çıkış denklemleri aşağıdaki gibidir [24]:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (3.13)$$

$$z_k = Hx_k + v_k \quad (3.14)$$

Denklemden belirtilen ve rastgele değişkenler olan w_k ve v_k değerleri sırasıyla proses ve ölçüm gürültüleri olarak adlandırılır. Bu parametreler birbirinden bağımsızdır ve aşağıda belirtilen normal olasılık dağılımlarına sahiptir:

$$p(w) \sim N(0, Q) \quad (3.14)$$

$$p(v) \sim N(0, R) \quad (3.15)$$

Denklemden $n \times n$ boyutlu olan A matrisi, bir önceki zaman adımındaki durum $(k-1)$ ile şimdiki zaman adımındaki durumu (k) ilişkilendirir. Pratikte A matrisi her bir zaman adımında değişebilir ancak genelde sabit kabul edilir. $n \times 1$ boyutlu olan B matrisi ise x 'e uygulanan kontrol girdisi $u \in R$ ile ilgilidir. $m \times n$ boyutlu olan H matrisi ise, sistemin x durumuyla z_k ölçümünü ilişkilendirir. Pratikte H değeri her bir zaman adımında değişse de genelde sabit kabul edilir.

$\hat{x}_k^- \in R^n$ değeri, k adımından önceki proses hakkında bilgilerin verildiği k adımından önceki durum tahminini oluşturur. $\hat{x}_k \in R^n$ ise, verilen z_k ölçümü ile birlikte k adımından sonraki durum tahmini oluşturur. Önceki ve sonraki durum tahminlerindeki hatalar aşağıdaki şekilde olur [24]:

$$e_k^- \equiv x_k - \hat{x}_k^- \quad (3.16)$$

$$e_k \equiv x_k - \hat{x}_k \quad (3.17)$$

Bu durumda önceki hata kovaryansı tahmini aşağıdaki gibidir [24]:

$$P_k^- = E[e_k^- e_k^{-T}] \quad (3.18)$$

Sonraki hata kovaryansı tahmini ise aşağıdaki gibidir [24]:

$$P_k = E[e_k e_k^T] \quad (3.19)$$

Kalman filtresi için denklemleri türetirken, ilk olarak, aşağıdaki denklem ele alınır [24]. Buna göre sonraki durum tahmini $\hat{x}_k$ değerinin, önceki durum tahmini olan $\hat{x}_k^-$ değeriyle ölçüm değeri olan z_k değeri arasında lineer bir ilişki kurulur. Denklemdaki $H\hat{x}_k^-$ değeri ise ölçüm tahmini olarak bilinir.

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \quad (3.20)$$

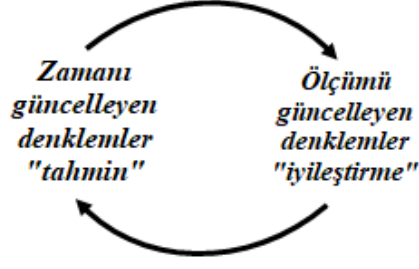
Denklemdaki $z_k - H\hat{x}_k^-$ değeri ölçüm kalanı (measurement residual) olarak da bilinir. Gerçek ölçüm değeri ile tahmin edilen ölçüm değeri arasındaki farkı yansıtır.

Kalman kazancı olarak da bilinen $n \times m$ boyutlu K matrisi, aşağıdaki denklemde ifade edilmiştir [24]:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} = \frac{P_k^- H^T}{H P_k^- H^T + R} \quad (3.21)$$

Kalman kazancını ifade eden denkleme bakıldığında, R ölçüm hatası kovaryansı değerinin sıfıra yaklaştığı durumda, z_k ölçüm değerinin daha güvenilir, $H\hat{x}_k^-$ değerinin ise daha az güvenilir olduğu görülür. Diğer taraftan, önceki hatanın tahmin kovaryansı değeri P_k^- sıfıra yaklaştıkça, z_k ölçüm değerinin daha az güvenilir, $H\hat{x}_k^-$ ölçüm tahmini değerinin daha fazla güvenilir olduğu görülür.

Kalman filtresi, geri bildirim (feedback) kontrol mekanizmasını kullanarak durum tahmini yapar. Filtre belirli bir zamandaki durum hakkında tahminde bulunur ve ardından buna göre ölçümler (gürültülü) şeklinde geri bildirimler elde eder. Kalman filtresi denklemleri iki gruba ayrılır: zamanı güncelleyen denklemler (time update equations) ve ölçümü güncelleyen denklemler (measurement update equations). Zamanı güncelleyen denklemler, bir sonraki zaman adımında önsel (priori) tahminleri elde etmek için, mevcut durum ve hata kovaryansını ileriye doğru yansıtmaktan sorumludur. Ölçümü güncelleyen denklemler ise geri bildirimden sorumludur. Önceki tahmin değerlerini kullanarak sonraki tahmin değerlerinin güncellenmesini sağlar. Zamanı güncelleyen denklemler tahminler üretirken, ölçümü güncelleyen denklemler bu tahminlerin geri bildirim mekanizması sayesinde iyileştirilmesinden sorumludur.



Şekil 3.10 : Kalman filtresi döngüsü.

Ayrık-Kalman filtresi (discrete Kalman filter) için zamanı güncelleyen denklemler, aşağıdaki gibidir [24]:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (3.22)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (3.23)$$

Bu denklemler durum ve kovaryans tahminlerini $(k - 1)$ adımından k adımına taşımakla sorumludur.

Ayrık-Kalman filtresi için ölçümü güncelleyen denklemler aşağıda verilmiştir: [24]

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (3.24)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (3.25)$$

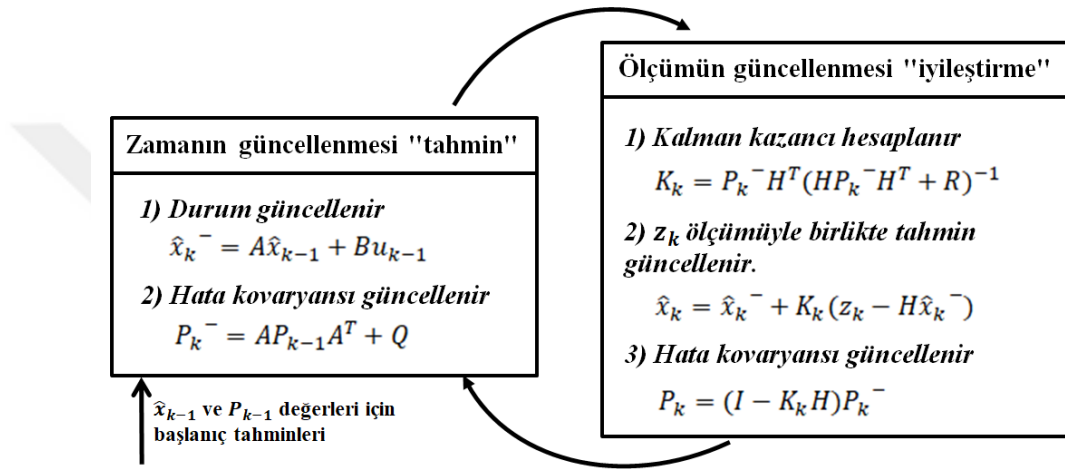
$$P_k = (I - K_k H)P_k^- \quad (3.26)$$

Ölçümü güncelleyen denklemleri ilk görevi K_k Kalman kazancını hesaplamaktır. Sonraki adım ise prosesin gerçek ölçümünü alarak z_k değerini elde edip, buna göre sonsal durum tahmini (posteriori state estimate) olan $\hat{x}_k$ değerini oluşturmaktır. Sonraki adımda ise sonsal hata kovaryansı (posteriori error covariance) P_k değeri elde edilir.

Filtre, gerçekte uygulamaya alınırken, ölçüm gürültüsü kovaryansı (measurement noise covariance) R 'nin değeri genelde önceden ölçülür. Zaten filtre çalışırken, prosesin ölçümünü almaya ihtiyaç duyduğumuz için, ölçüm hatası kovaryansı olan R değerinin ölçülmesi mümkündür. Bu yüzden, ölçüm gürültüsü varyansını belirlemek için filtre çalıştırılmadan ve prosesin dışında da olsa bazı ölçüm örneklerini alırız.

Proses gürültüsü kovaryansı (process noise covariance) Q değerinin belirlenmesi, tahmin etmeye çalıştığımız prosesi direkt olarak gözlemleyemeyeceğimiz için daha zordur. Bazen, göreceli olarak daha basit olan bir proses modeli, uygun Q seçimiyle prosese yeteri miktarda belirsizlik verilmesi durumunda, kabul edilebilir sonuçlar doğurabilir. Bu durumda da, proses ölçümlerinin güvenilir olduğu varsayımı yapılır.

İki durumda da, bu parametreleri seçerken mantıklı ve kabul edilebilir bir temele sahip olunmalıdır. Çoğu zaman, filtrenin düzgün bir biçimde çalışması, Q ve R değerlerinin düzgün bir şekilde ayarlanmasıyla ilgilidir [24].



Şekil 3.11 : Kalman filtresinin bütünü gösteren diyagram [24].

3.4 Genişletilmiş Kalman Filtresi

Önceki bölümlerde bahsedildiği gibi, Kalman filtresi, lineer stokastik fark denklemi tarafından tanımlanan ayrık-zaman kontrollü bir prosesin $x \in R^n$ durumunu tahmin etmeye çalışan genel bir problemin çözümünde kullanılır. Fakat tahmin edilecek proses veya prosesin ölçümle olan ilişkisinin lineer olmadığı (nonlinear) durumlarda ne olacağı bilinmemektedir. Böyle durumlar için Kalman filtresinin son derece yararlı bir uygulaması bulunmaktadır. Lineerleştirme işlemini uygulayan bu Kalman filtresi, genişletilmiş Kalman filtresi (extended Kalman filter) olarak bilinmektedir.

Bu kapsamda, proses ve ölçüm fonksiyonlarının kısmi türevleri kullanılarak, mevcut tahmin değeri etrafında lineerleştirme işlemini gerçekleştirebiliriz. Prosesin, lineer olmayan fark denklemiyle tanımlanan bir $x \in R^n$ durum vektörü ile temsil edildiğini düşünelim. $z \in R^m$, ölçüm değerini gösterir ve denklemler aşağıdaki gibi olur [24]:

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \quad (3.27)$$

$$z_k = h(x_k, v_k) \quad (3.28)$$

Denklemdaki w_k ve v_k değerleri sırasıyla proses ve ölçüm gürültülerini temsil etmektedir. Lineer olmayan f fonksiyonu, $(k-1)$ adımındaki durum ile k adımındaki durumu ilişkilendiren bir fark denklemdir. Aynı zamanda parametre olarak, u_{k-1} girdi (input) fonksiyonunu ve w_k proses gürültüsünü de (process noise) içerir. Lineer olmayan h fonksiyonu ise, x_k durumuyla z_k ölçümünü ilişkilendirir.

Pratikte, her bir zaman adımı için w_k ve v_k değerleri bilinmemektedir. Fakat bu değerler olmadan aşağıdaki şekilde durum ve ölçüm vektörleri için yaklaşık değerler hesaplanabilir [24]:

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (3.29)$$

$$\tilde{z}_k = h(\tilde{x}_k, 0) \quad (3.30)$$

Denklemlerdeki $\hat{x}_k$ değeri, sonraki durum tahmin değerini gösterir.

Genişletilmiş Kalman filtresinin temel eksikliği, sistemdeki rastgele değişkenlerin lineer olmayan dönüşümler geçirdikten dağılımlarının artık normal dağılım olarak kalmamasıdır.

Lineer olmayan durum ve ölçüm ilişkilerine sahip bir proses için tahminde bulunurken, ilk olarak tahmini lineerize eden (doğrusallaştıran) aşağıdaki yeni denklemlerin yazılmasıyla başlanır [24]:

$$x_k \approx \tilde{x}_k + A(x_{k-1} - \tilde{x}_{k-1}) + Ww_{k-1} \quad (3.31)$$

$$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + Vv_k \quad (3.32)$$

İlgili denklemler için parametreler aşağıdaki şekilde açıklanmıştır [24]:

- x_k ve z_k sırasıyla gerçek durum (actual state) ve ölçüm vektörleridir.
- $\tilde{x}_k$ ve $\tilde{z}_k$ sırasıyla yaklaşık durum (approximate state) ve ölçüm vektörleridir.
- $\hat{x}_k$, k adımındaki durumun sonsal tahminidir. (posteriori estimate)
- w_k ve v_k rastgele değişkenleri, sırasıyla proses ve ölçüm gürültülerini oluşturur.

- A Jacobian matrisi, f fonksiyonunun x 'e göre alınan kısmi türevlerinden oluşur [24].

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0) \quad (3.33)$$

- W matrisi, f fonksiyonunun w 'e göre alınan kısmi türevlerinden oluşur [24].

$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0) \quad (3.34)$$

- H matrisi, h fonksiyonunun x 'e göre alınan kısmi türevlerinden oluşur [24].

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\tilde{x}_k, 0) \quad (3.35)$$

- V matrisi, h fonksiyonunun v 'e göre alınan kısmi türevlerinden oluşur [24].

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\tilde{x}_k, 0) \quad (3.36)$$

A , W , H ve V Jacobian matrisleri her bir zaman adımında değişmesine rağmen, gösterim kolaylığı açısından hangi zaman adımında olunduğunu gösteren k alt indisi kullanılmamıştır.

Genişletilmiş Kalman filtresi için denklem takımları aşağıdaki gibidir. Buna göre öncül (priori) kavramını gösterebilmek için, $\tilde{x}_k$ yerine, $\hat{x}_k^-$ kullanılır. Aynı zamanda A , W , H ve V Jacobian matrisleri her bir zaman adımında değiştiği için k altindisi de bunlara eklenir. Genişletilmiş Kalman filtresi için zamanı güncelleyen denklemler (time update equations) aşağıdaki gibidir [24]:

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (3.37)$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \quad (3.38)$$

Ayrık zaman (discrete time) Kalman filtresinde olduğu gibi durum ve kovaryans tahminleri önceki yani $(k - 1)$ zaman adımından, şimdiki zaman adımı olan k 'ya taşınır. Denklemdeki A_k ve W_k , k adımındaki proses Jacobian matrisleridir. Q_k ise, k adımındaki proses gürültü kovaryansıdır. (process noise covariance)

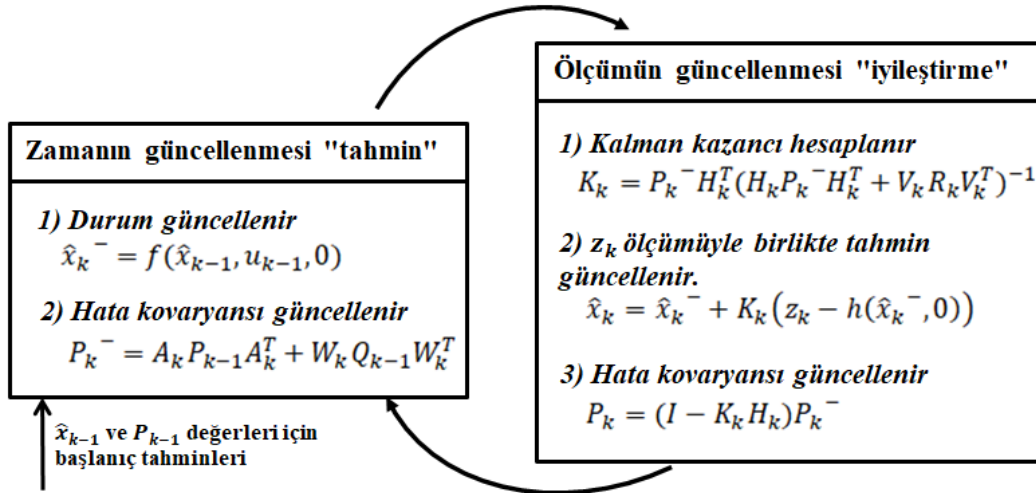
Genişletilmiş Kalman filtresi için ölçümü güncelleyen denklemler (measurement update equations) gibidir [24]:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \quad (3.39)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0)) \quad (3.40)$$

$$P_k = (I - K_k H_k) P_k^- \quad (3.41)$$

Ayrık zaman (discrete time) Kalman filtresinde olduğu gibi, ölçümü güncelleyen denklemler, z_k ölçümünü de kullanılarak, durum ve kovaryans tahminlerini iyileştirir. Denklemdaki H_k ve V_k , k adımıdaki ölçüm Jacobian matrisleridir. R_k ise, k adımıdaki ölçüm gürültü kovaryansıdır (measurement noise covariance). Genişletilmiş Kalman filtresinde kullanılan denklemler aşağıdaki şekilde gösterilmiştir [24].



Şekil 3.12 : Genişletilmiş Kalman filtresi denklemleri [24].

Kalman filtresi, çeşitli proseslerin değişkenlerini veya durum bilgilerini tahmin edebilen bir araçtır. Matematiksel açıdan, Kalman filtresinin bir sistemin durumlarını tahmin ettiği söylenebilir. Bir sistemin durumlarını bilmeye çalışmamızın iki nedeni vardır [31]:

- İlk olarak, sistemi kontrol edebilmek için durumlarını tahmin etmeye ihtiyaç duyarız. Örneğin, elektrik mühendisleri, motorun pozisyonunu kontrol edebilmek için motor sarımlarındaki akımları tahmin etmek durumundadır. Uzun

mühendisleri ise, bir uydunun yörüngesini kontrol edebilmek için, uydunun hızını tahmin etmeye çalışır.

- İkinci olarak, sistem durumları, sistemi oluşturan parametreler hakkında bilgi sahibi olmaya çalıştığımız için tahmin edilmeye çalışılır. Örneğin, elektrik mühendisleri, potansiyel hataları öngörebilmek için güç sistem parametrelerini tahmin etmeye çalışır. Uzay mühendisi, bir uydunun gelecekteki etkinliklerini planlayabilmek için uydu konumunu tahmin etmelidir.

Standart Kalman filtresi tahmin yapabilmek için etkili bir araç olsa da lineer sistemlerle sınırlıdır. Gerçek dünyadaki çoğu uygulama lineer olmadığından, Kalman filtresi direkt olarak uygulanamaz. Gerçek dünyada daha çok lineer olmayan filtreler kullanılır.

Eğer bir sinyali standart Kalman filtresi ile tahmin etmeye çalışıyorsak, ölçtüğümüz sistem lineer sistem denklemleriyle tanımlanabilmelidir. Bir lineer sistem prosesi aşağıdaki iki denklemle açıklanabilir [31]:

- Durum denklemi:

$$x_{k+1} = Ax_k + Bu_k + w_k \quad (3.42)$$

- Çıkış denklemi:

$$y_k = Cx_k + v_k \quad (3.43)$$

Üstteki denklemlere ait bazı parametreler aşağıdaki gibidir:

- A , B ve C matristir.
- k , zaman indeksidir.
- x , sistemin durum vektörü olarak adlandırılır.
- u , değeri bilinen bir kontrol girdisidir. (kontrol sinyali olarak da adlandırılır.)
- y , ölçülen çıkış değeridir.
- w , proses gürültüsü ve v , ise ölçüm gürültüsü olarak adlandırılır.

Durum tahmini problemlerinde, sistem hakkında bütün bilgiyi içerdiği için x durumu tahmin edilmeye çalışılır. x değeri direkt olarak ölçülebilen bir parametre değildir.

Bunun yerine, x 'in bir fonksiyonu olan ve v gürültüsü tarafından bozulan, y ölçülür. x ile ilgili bir tahmin değeri elde edebilmek için y değeri kullanılır.

Örneğin, bizim sistemimiz, düzgün bir çizgide hareket eden herhangi bir araç olsun. Bu durumda, araç için durum bilgilerinin pozisyon ve hız bilgisinden oluştuğunu söyleyebiliriz. u girdisi ivme ve y çıkışı ise ölçülen pozisyon değeridir. Her T saniyesinde pozisyon bilgisinin ölçüldüğünü varsayalım. Bu sistem, aşağıdaki gibi modellenir [31]:

$$x_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} T^2 \\ T \end{bmatrix} u_k + w_k \quad (3.44)$$

$$y_k = [1 \quad 0]x_k + v_k \quad (3.45)$$

Bu denklemlerde, x_k , aracın k anındaki hız ve pozisyon bilgilerini bir vektördür. u_k , ivmeyi gösteren skaler bir değerdir. y_k ölçülen pozisyon değerini gösteren bir skalerdir. w_k ve v_k değerleri ise sırasıyla proses ve ölçüm gürültüleridir.

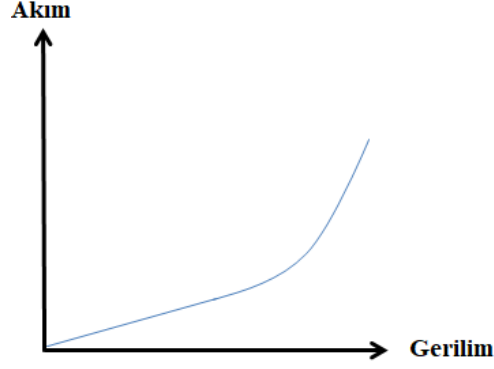
Aracın pozisyonunu belirli bir yol izlemesini istediğimiz için kontrol etmek istediğimizi varsayalım. Pozisyon tahmini için sadece y_k değerini kullanabiliriz fakat y_k değeri de gürültülüdür. Bu durumda Kalman filtresini kullanmamız iyi bir yöntem olabilir. Çünkü Kalman filtresi sadece pozisyon ölçümü olan y_k değerini değil, aynı zamanda durum denklemlerindeki diğer parametrelere ait bilgileri de kullanır. Kalman filtresi denklemleri aşağıdaki şekilde yazılabilir [31]:

$$K_k = P_k C^T (C P_k C^T + R)^{-1} \quad (3.46)$$

$$\hat{x}_{k+1} = (A\hat{x}_k + Bu_k) + K_k(y_k - C\hat{x}_k) \quad (3.47)$$

$$P_{k+1} = A(I - K_k C)P_k A^T + Q \quad (3.48)$$

Kalman filtresi lineer sistemlere uygulanabilen lineer bir filtredir. Fakat lineer sistemler gerçek dünyada bulunmazlar. Bütün sistemler eninde sonunda lineer değildir. Hatta Ohm kanununu ($I = V/R$) düşündüğümüzde bile, bu ilişkinin sadece belirli bir aralıkta geçerli olduğunu görürüz. Eğer bir dirence uygulanan gerilim değeri belirli bir değeri aşarsa, Ohm kanunu uygulanamaz. Şekil 3.13'te bu ilişki açıklanmıştır.



Şekil 3.13 : Lineer olmayan akım/gerilim ilişkisi.

Çoğu sistem lineer sistemlere benzerlik gösterebilir ve bu sayede lineer yaklaşımlar iyi sonuçlar verebilir. Bu sonuçlar sistem durumlarını tahmin etmemiz için belirli bir noktaya kadar yardımcı olabilir. Bu yüzden lineer olmayan filtre yöntemlerine başvurmak zorunda kalırız.

Lineer olmayan filtreleme zor ve karmaşık olabilir. Lineer olmayan filtreleme lineer filtreleme kadar iyi anlaşılamamıştır. Fakat, bazı lineer olmayan filtreleme yöntemlerinin gittikçe yaygınlaşmaya başladıkları görülmüştür. Bu yöntemlerin içinde, Kalman filtresinin lineer olmayan uzantıları, kokusuz (unscented) filtre ve parçacık (particle) filtresi bulunmaktadır [31].

Lineer olmayan Kalman filtresinde önemli olan nokta, sistemdeki lineer olmayan terimleri nominal bir $\bar{x}$ noktası etrafında olmak üzere Taylor açılımı şeklinde yazmaktır. Lineer olmayan bir fonksiyonun Taylor serisindeki yazılışı, [31]'e göre, aşağıdaki şekildedir [31]:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(\bar{x})\Delta x^n}{n!} \quad (3.49)$$

Yukarıdaki denklemlerde yer alan bazı parametreler aşağıda verilmiştir:

- $\Delta x = x - \bar{x}$
- $f^{(n)}(\bar{x})$, $f(x)$ 'in n 'inci türevidir. ($x = \bar{x}$)

Bir fonksiyonun lineerize edilmesi, belirli bir nokta etrafında birinci derecede Taylor serisi açılımının uygulanması anlamına gelmektedir. Bir $f(x)$ fonksiyonu için birinci dereceden Taylor açılımının uygulanması aşağıdaki gibidir:

$$f(x) = f(\bar{x}) + f'(\bar{x})\Delta x \quad (3.50)$$

Taylor serisi açılımından anladığımız kadarıyla doğrusallaştırılmış Kalman filtresi (linearized Kalman filter) için ilgili denklilikleri türetebiliriz. Doğrusallaş edilmiş Kalman filtresinin temel mantığı, lineer olmayan bir sistemle başlayarak, daha sonrasında lineer olmayan sistemin nominal yörüngesinden (nominal trajectory) sapmaları temsil eden lineer bir sistem bulmaktır. Bu aşamadan sonra, nominal yörüngeden sapmaları tahmin edebilmek için Kalman filtresini kullanabiliriz. Bu bize dolaylı olarak da olsa lineer olmayan sistem için durum tahminlerini verir.

Doğrusal olmayan bir sistemin genel modeli aşağıdaki gibidir:

- Durum denklemi (state equation):

$$x_{k+1} = f(x_k, u_k) + w_k \quad (3.51)$$

- Çıkış denklemi (output equation):

$$y_k = h(x_k) + v_k \quad (3.52)$$

$f(\cdot)$ durum denklemi ve $h(\cdot)$ ölçüm denklemi lineer olmayan fonksiyonlardır.

Eğer durum veya çıkış denklemi lineer olmayan terimlere sahipse sistem lineer değildir.

Lineerize edilmiş Kalman filtresinde, durum ve çıkış denklemi için nominal bir durum etrafında birinci dereceden Taylor serisi açılımı uygulanır. Nominal durum, zamanın bir fonksiyonu olduğundan bazen yörünge (trajectory) olarak da isimlendirilir. Nominal yörünge, sistem davranışının nasıl olacağına dair bir tahmine dayanır. Örneğin sistem denklemleri bir uçağın dinamiklerini temsil ederse, nominal durum planlanmış bir uçuş yörüngesi olabilir. Gerçekte olan uçuş yörüngesi, modeldeki hatalar, bozucu girdiler veya görülmeyen bazı etkenlerden dolayı nominal yörüngeden farklı olacaktır. Fakat gerçek yörünge nominal yörüngeye yeteri kadar yakın olduğundan Taylor serisi lineerleştirmesi düzgün bir biçimde uygulanabilecektir. Lineer olmayan bir sistem için Taylor serisi lineerleştirmesinde kullanılan denklemler aşağıdaki gibidir [31]:

$$x_{k+1} = f(x_k, u_k) + w_k \approx f(\bar{x}_k, u_k) + f'(\bar{x}_k, u_k)\Delta x_k + w_k \quad (3.53)$$

$$y_k = h(x_k) + v_k \approx h(\bar{x}_k) + h'(\bar{x}_k)\Delta x_k + v_k \quad (3.54)$$

Nominal yörüngedeki sapmalar aşağıdaki şekilde yazılabilir [31]:

$$\Delta x_{k+1} = x_{k+1} - \bar{x}_{k+1} = x_{k+1} - f(\bar{x}_k, u_k) \quad (3.55)$$

$$\Delta y_k = y_k - \bar{y}_k = y_k - h(\bar{x}_k) \quad (3.56)$$

Daha önce verilen denklemlerle bu denklemleri bir araya getirdiğimizde aşağıdaki sonuçları elde ederiz [31]:

$$\Delta x_{k+1} = f'(\bar{x}_k, u_k)\Delta x_k + w_k \quad (3.57)$$

$$\Delta y_k = h'(\bar{x}_k)\Delta x_k + v_k \quad (3.58)$$

Δx ve Δy fonksiyonları ile lineer olan durum (state) ve çıkış (output) denklemleri elde edilir. Bu durumda Δx değerini tahmin edebilmek için standart Kalman filtresini kullanabiliriz.

Lineerize edilmiş Kalman filtresini kullanırken dikkat etmemiz gereken iki nokta bulunmaktadır:

- Δx değerini tahmin edebilmek için standart Kalman filtresini kullandıktan sonra, x durum tahminini elde edebilmek için, Δx tahmin değerine nominal durum (state) $\bar{x}$ değerini de eklememiz gerekir. ($\Delta x = x - \bar{x}$)
- Eğer x gerçek durum bilgisi, nominal durum $\bar{x}$ değerinden çok uzak ise, lineerize edilmiş Kalman filtresi düzgün sonuçlar vermeyecektir.

Lineerize edilmiş Kalman filtresi algoritmasının özeti aşağıdaki gibidir [31]:

- Sistem denklemleri aşağıdaki gibidir:

Durum denklemi:

$$x_{k+1} = f(x_k, u_k) + w_k \quad (3.59)$$

Çıkış denklemi:

$$y_k = h(x_k) + v_k \quad (3.60)$$

- Nominal yörünge denklemleri aşağıdaki gibidir [31]:

$$\bar{x}_{k+1} = f(\bar{x}_k, u_k) \quad (3.61)$$

$$\bar{y}_k = h(\bar{x}_k) \quad (3.62)$$

- Her bir adımda aşağıdaki kısmi türev matrisleri hesaplanır [31]:

$$A_k = f'(\bar{x}_k, u_k) \quad (3.63)$$

$$C_k = h'(\bar{x}_k) \quad (3.64)$$

Buradaki türevler x_k parametresine göre alınır.

- y_k gerçek ölçüm değeri ile $\bar{y}_k$ nominal ölçüm değeri arasındaki farkı gösteren Δy_k değer tanımlanır [31]:

$$\Delta y_k = y_k - \bar{y}_k = y_k - h(\bar{x}_k) \quad (3.65)$$

- Aşağıdaki Kalman filtresi denklemleri uygulanır [31]:

$$K_k = P_k C_k^T (C_k P_k C_k^T + R)^{-1} \quad (3.66)$$

$$\Delta \hat{x}_{k+1} = A_k \Delta \hat{x}_k + K_k (\Delta y_k - C_k \Delta \hat{x}_k) \quad (3.67)$$

$$P_{k+1} = A_k (I - K_k C_k) P_k A_k^T + Q \quad (3.68)$$

$$\hat{x}_{k+1} = \bar{x}_{k+1} + \Delta \hat{x}_{k+1} \quad (3.69)$$

Lineerize edilmiş Kalman filtresine ilişkin denklemler yukarıdaki gibidir. Bu denklemlerde de görüldüğü gibi bu filtre yöntemini uygularken karşılaşılan problem $\bar{x}$ nominal yörünge (nominal trajectory) bilgisinin önceden bilinmesi durumudur. Bazı sistemlerde bu bilgi önceden bilinse de çoğu sistemde bu bilgiyi önceden edinmenin bir yolu yoktur.

Genişletilmiş Kalman filtresinin temel mantığı, x tahmin değerini, lineerize edilmiş Kalman filtresindeki nominal yörünge değeri olarak kullanmaktır. Başka bir deyişle, lineerleştirilmiş Kalman filtresindeki $\bar{x}$ değeri, $\hat{x}$ olarak ayarlanır. x değerini tahmin etmek için nominal yörüngeyi kullanıp, tahmin edilen değeri de nominal yörünge olarak kullanırız. Bu değişikliklerin lineerleştirilmiş Kalman filtresine

yerleştirilmesinin ardından, genişletilmiş Kalman filtresi algoritmasını aşağıdaki şekilde elde ederiz:

- Sistem denklemleri aşağıdaki gibidir [31]:

Durum denklemi:

$$x_{k+1} = f(x_k, u_k) + w_k \quad (3.70)$$

Çıkış denklemi:

$$y_k = h(x_k) + v_k \quad (3.71)$$

- Her bir adımda aşağıdaki kısmi türev matrisleri hesaplanır [31]:

$$A_k = f'(\hat{x}_k, u_k) \quad (3.72)$$

$$C_k = h'(\hat{x}_k) \quad (3.73)$$

Buradaki türevler x_k parametresine göre alınır. Sonrasında ise $x_k = \hat{x}_k$ olarak alınır.

- Aşağıdaki Kalman filtresi denklemleri uygulanır [31]:

$$K_k = P_k C_k^T (C_k P_k C_k^T + R)^{-1} \quad (3.74)$$

$$\hat{x}_{k+1} = f(\hat{x}_k, u_k) + K_k [(y_k - h(\hat{x}_k))] \quad (3.75)$$

$$P_{k+1} = A_k (I - K_k C_k) P_k A_k^T + Q \quad (3.76)$$



4. ÇOK ETMENLİ SİSTEMLERDE KALMAN FİLTRESİ UYGULAMALARI

Bu bölümde, her biri birbirinden farklı dinamiklere sahip olmak üzere, çok etmenli sistemler ele alınacaktır. Söz konusu sistemler belirli sayıda elemanlara veya üyelere sahiptir. Sistemdeki her bir eleman belirli parametrelere sahiptir. Bu parametreleri elemanların sahip oldukları bilgiler olarak düşünebiliriz. Her bir eleman sahip olduğu bu bilgiyi sistemdeki diğer komşu elemanlarla paylaşarak belirli bir değerde veya durum bilgisinde anlaşmaya diğer bir deyişle iş birliğine (consensus) varmaya çalışacaktır.

Çok etmenli sistemlerdeki elemanlar, belirli komşuluk ilişkisine ve belirli rastlantısal oran kapsamında iletişime geçerek, sahip oldukları bilgileri paylaşacaktır. Sistemlerin iletişime geçebilme ihtimalleri yani komşu olabilmeleri, yakınlıklarına bağlı olarak rastlantısal bir biçimde belirlenecektir.

Ele alınan her bir sistem belirli bir kontrol protokolüne sahiptir. Bu kontrol protokolü için gerekli parametreler gradyant düşümü yoluyla güncellenen adaptif PD yardımıyla iteratif bir şekilde ayarlanacaktır.

Sistemlerdeki elemanlar belirli bir değer veya durum bilgisinde iş birliğine varmaktadır. Fakat yapılan çalışma kapsamında söz konusu sistemlerde bulunan elemanların sahip oldukları bilgilere gürültü eklenerek, haberleşmelerinin düzgün bir biçimde olması yerine gürültülü olması sağlanmıştır. Bu gürültü nedeniyle elemanların diğer komşu elemanlardan aldıkları ve diğer komşu elemanlara verdikleri bilgi gürültülü hale getirilmiştir. Sistemi oluşturulan elemanların bilgi transferleri gürültülü hale getirildikten sonra Kalman filtresinin çeşitli uygulamaları gerçekleştirilerek, sistemi oluşturan elemanlara daha doğru ve gürültüsü bastırılmış bilgilerin iletilmesi sağlanmıştır. Bu sayede iş birliğinin bozucu girdilere rağmen düzgün bir biçimde gerçekleştirilmesi sağlanmıştır. Söz konusu sistemlerin davranışlarını gözlemlemek adına MATLAB ortamında kodlama gerçekleştirilmiştir.

4.1 Çift Entegrator Sistemi

Bu bölümde çok etmenli bir sistem olan çift entegrator PD sistemi ele alınacaktır. Söz konusu sistem belirli bir sayıda elemandan oluşmaktadır. Bu elemanlar arasında bilgi alışverişi özellikleri daha önce de belirtilen bir ağ topolojisi üzerinden sağlanmakta ve anlaşılan bir değer veya durum bilgisi üzerinde iş birliği sağlanmaktadır.

Mevcut sisteme ve sistemdeki elemanların bilgilerine gürültü eklenip, bozucu girdiler sisteme dahil edilmiştir. Söz konusu sistem lineer olduğu için temel Kalman filtresi denklemleri kullanılarak sistemin bu gürültülü ortamda bile sağlıklı bir şekilde iş birliğine varması sağlanmıştır.

4.1.1 Sisteme ait parametre ve denklemler

Çok etmenli sisteme bir örnek olarak verilen çift integrator sistemi belirli parametrelerden oluşmaktadır. İlerleyen kısımlarda bu parametreler ve birbirleriyle olan ilişkileri belirten denklemlerden bahsedilecektir.

Çift entegratör sistemi için temel kontrol ve iş birliği protokolü aşağıdaki denklem çerçevesinde sağlanacaktır:

$$\ddot{x}_i(t) = - \sum_{j=1}^n a_{ij} (x_i(t) - x_j(t)) - \beta \sum_{j=1}^n a_{ij} (\dot{x}_i(t) - \dot{x}_j(t)) \quad (4.1)$$

Denkleme göre, x ve $\dot{x}$ olmak üzere elemanların sahip oldukları bilgiler belirli a_{ij} komşuluk ilişkisine göre belirlenmektedir.

Çift entegratör sistemini oluşturan belirli sayıda eleman vardır. Bu elemanları tanımlamak için etmen (agents) ifadesi de kullanılmaktadır. Sistemi oluşturan elemanlar belirli bilgilere sahiptir ve bu bilgiler bütün iterasyonlar için başlangıçta rastlantısal bir şekilde atanmıştır.

Şekil 4.1’de gösterildiği gibi her bir elemanın " x " ve " $\dot{x}$ " değişkenleri oluşturulmuştur. Şekil 4.1’de gösterilen “agents” değişkeni eleman sayısını, “Iterations” değişkeni ise iterasyon sayısını göstermektedir. Şekil 4.1’e göre belirlenen " x " ve " $\dot{x}$ " değişkenleri ölçüm değişkenlerine atanmak üzere sırasıyla " $x_{measured}$ " ve " $\dot{x}_{measured}$ " değerlerine atanmıştır.

```

for t=1:Niterations
    for i=1:agents
        a=rand(1);
        if a >= 0.5
            b=1;
        else a < 0.5
            b=-1;
        end
        x(t,i)=1*b*rand(1);
        a=rand(1);
        if a >= 0.5
            b=1;
        else a < 0.5
            b=-1;
        end
        xdot(t,i)=1*b*rand(1);
    end
    xmeasured(t,:)=x(t,:);
    xdotmeasured(t,:)=xdot(t,:);
end

```

Şekil 4.1 : Sistemdeki elemanların bilgilerinin oluşturulması

Daha önceki bölümlerde anlatıldığı gibi çok etmenli sistemlerde bulunan elemanlar birbirlerine belirli bir ağ topolojisine göre bağlıdır. Bu ağ topolojisinde bilgi transferi graf teorisinde de belirtildiği gibi komşuluk ilişkisine dayanır.

```

for i=1:agents
    xlocation(i)=100*rand(1);
    ylocation(i)=100*rand(1);
end

for i=1:agents
    for j=1:agents
        A(i,j)=0;
    end
end

for i=1:agents
    for j=1:agents
        distance(i,j)=sqrt((xlocation(i)-xlocation(j))^2+(ylocation(i)-ylocation(j))^2);
        if distance(i,j) <= r_threshold
            A(i,j)=1;
        end
    end
end
end

```

Şekil 4.2 : Sistemdeki elemanların komşuluk ilişkilerinin oluşturulması

Şekil 4.2’de sistemi oluşturan elemanlar için belirli konum bilgileri rastlantısal olacak şekilde oluşturulmaktadır. Buna göre her bir eleman için x ve y koordinatlarındaki pozisyonlarını belirlemek üzere sırasıyla " $xlocation$ " ve " $ylocation$ " değişkenleri oluşturulur.

Şekil 4.2’de belirtildiği gibi elemanlar arasındaki komşuluk aralarındaki mesafeye göre belirlenmektedir. Buna göre aralarındaki mesafeler hesaplanan elemanlar, bu

mesafenin belirli bir değerin altında kalması durumunda komşudurlar ve ilgili "A" matrisinin ilgili hücrelerine 1 değeri atanır.

Söz konusu çok etmenli sistemde iş birliği protokolü katsayıları gradyant düşümü yöntemi ile ayarlanan bir adaptif PD sayesinde sağlanır ve bu bütün elemanlar için bir kurallar bütünü olarak dışarıdan alınan bir girdi olarak kabul edilir.

Gradyant düşümü denklemlerine göre adaptif PD’de kullanılan " K_p " ve " K_d " parametreleri ayarlanır. Gradyant düşümü yönteminde bedel fonksiyonu (cost function) aşağıdaki gibidir:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \quad (4.2)$$

Denklem 4.1’deki " $h_{\theta}(x^{(i)}) - y^{(i)}$ " terimi Şekil 4.3’de verilen "*term1*" ve "*term2*" terimlerini temsil eder. Buna göre PD katsayıları aşağıdaki denkleme göre güncellenir:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (4.3)$$

```

for i=1:agents
    sum(t,i)=0;
    for j=1:agents
        if t == 1
            error(t,i,j)=xmeasured(t,i)-xmeasured(t,j);
            errordot(t,i,j)=xdotmeasured(t,i)-xdotmeasured(t,j);
        end
        if t > 1
            error(t,i,j)=xmeasured(t-1,i)-xmeasured(t-1,j);
            errordot(t,i,j)=xdotmeasured(t-1,i)-xdotmeasured(t-1,j);
        end
        term1=(error(t,i,j))*error(t,i,j);
        term2=errordot(t,i,j)*errordot(t,i,j);
        if t>1
            Kp(t,i,j)=Kp(t-1,i,j)-eta1*term1;
            Kd(t,i,j)=Kd(t-1,i,j)-eta2*term2;
        end
        sum(t,i)=sum(t,i)+Kp(t,i,j)*A(i,j)*error(t,i,j)+
            Kd(t,i,j)*A(i,j)*errordot(t,i,j);
    end
end

```

Şekil 4.3 : Sistemde bulunan elemanlar için kontrol girdilerinin oluşturulması.

Denklem 4.3’deki " θ_j " terimi baz alınarak aşağıda verilen adaptif PD denkleminde kullanılmak üzere " K_p " ve " K_d " değerleri güncellenir. Denklem 4.3’deki " α " terimi ise (öğrenme hızı katsayısı) Şekil 4.3’de verilen "*eta1*" ve "*eta2*" ile temsil edilir ve

bu değerlere programın başında sabit bir değer atanır. Aşağıda programda uygulanan temel adaptif PD denklemi verimiştir.

$$u = K_p(x_i - x_j) + K_d(\dot{x}_i - \dot{x}_j) \quad (4.4)$$

Söz konusu sistemdeki kontrol girdisi her bir eleman için her bir iterasyonda güncellenecek şekilde oluşturulur ve Şekil 4.3'de gösterilen "sum" değişkeni ile belirtilir ve denklem 4.3'de belirtilen kontrol girdisi veya iş birliği protokolü elde edilir.

Sisteme proses ve ölçüm gürültüsü eklenerek sistemin davranışı gözlenir. Buna göre daha önceki bölümlerde de bahsedilen proses ve ölçüm gürültü kovaryansı parametrelerine göre (Q ve R) sisteme normal dağılıma sahip " w " ve " v " bozucu girdileri eklenir.

```
xddotdot(t,:)=sum(t,:)+w(t,:);
xdot(t+1,:)=xdot(t,:)+xddotdot(t,:)*dT;
x(t+2,:)=x(t+1,:)+xdot(t+1,:)*dT;

Xvector1(t,:)=x(t+2,:);
Xvector2(t,:)=xdot(t+1,:);
Xvector(t,:)= [Xvector1(t,:) Xvector2(t,:)];

z(t,:)=Xvector(t,:)+[v(t,:) v(t,:)];
```

Şekil 4.4 : Sisteme gürültülerin eklenmesi.

```
%xmeasured_estimate
if t==1
    xpri(t,:)=b.*xpost(t,:);
    ppri(:, :, t)=AA*ppost(:, :, t)*AA'+Q*(eye(agents));
end
if t>1
    xpri(t,:)=b.*xpost(t-1,:);
    ppri(:, :, t)=AA*ppost(:, :, t-1)*AA'+Q*(eye(agents));
    kk(:, :, t)=ppri(:, :, t)/(ppri(:, :, t)+R*eye(agents));
    xpost(t,:)=xpri(t,:)+(kk(:, :, t)*
    ((xmeasured(t,:)-xpri(t,:)))');
    ppost(:, :, t)=(eye(agents)-kk(:, :, t))*ppri(:, :, t);

%xdotmeasured_estimate
if t==1
    xpri2(t,:)=b.*xpost2(t,:);
    ppri2(:, :, t)=AA*ppost2(:, :, t)*AA'+Q*(eye(agents));
end
if t>1
    xpri2(t,:)=b.*xpost2(t-1,:);
    ppri2(:, :, t)=AA*ppost2(:, :, t-1)*AA'+Q*(eye(agents));
    kk2(:, :, t)=ppri2(:, :, t)/(ppri2(:, :, t)+R*eye(agents));
    xpost2(t,:)=xpri2(t,:)+(kk2(:, :, t)*
    ((xdotmeasured(t,:)-xpri2(t,:)))');
    ppost2(:, :, t)=(eye(agents)-kk2(:, :, t))*ppri2(:, :, t);

xmeasured(t,:)=xpost(t,:);
xdotmeasured(t,:)=xpost2(t,:);
```

Şekil 4.5 : Kalman filtresi uygulanması

Şekil 4.5'e göre sisteme bozucu girdilerin eklenmesiyle birlikte, sistemi oluşturan bilgilerin iş birliğine düzgün bir şekilde varabilmeleri için Kalman filtresi uygulanır ve buna bağlı tahminler geliştirilerek sistemi oluşturan elemanların bilgilerine geri besleme yapılır.

Sonuç olarak bir sonraki bölümde detaylı bir şekilde gösterilecek olan grafikler elde edilir.

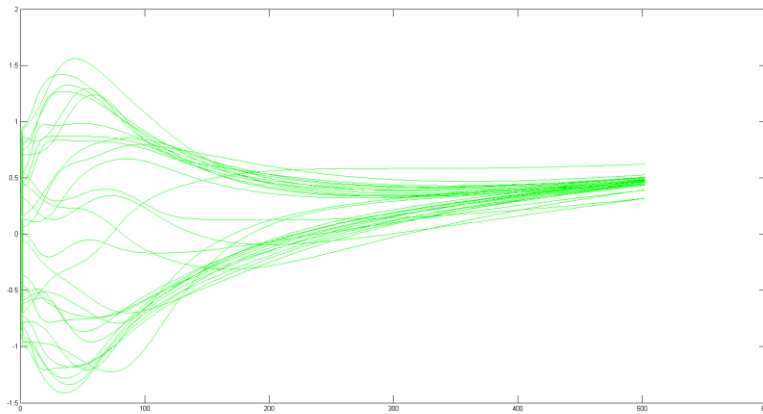
4.1.2 Sisteme ait simülasyon sonuçları ve sonuçların yorumlanması

Çok etmenli sistemlere örnek olarak çift entegratör sistemi incelenmiş ve ağ topolojisine sahip elemanların iş birliğine nasıl vardıkları üzerinde durulmuştur. Aynı zamanda sisteme ait parametrelere gürültü eklenerek bozucu girdilerin sistem davranışına nasıl etki ettiği gösterilmiştir.

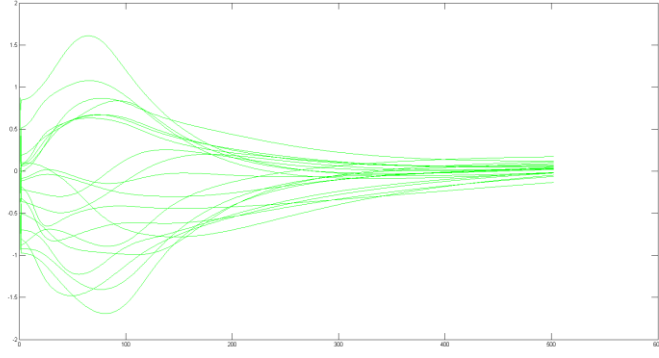
Bu bölümde sistemin davranışının çeşitli parametre değerlerinin değiştirilmesiyle nasıl değiştiği grafiksel olarak gösterilerek, söz konusu grafikler yorumlanacaktır.

İlk olarak gürültüsüz ortamda eleman (agents) sayısı değiştirilerek sonuçlar karşılaştırılacaktır.

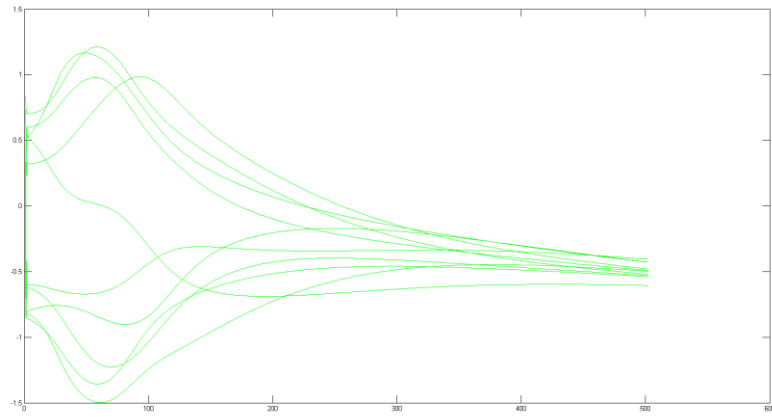
Şekil 4.6, Şekil 4.7 ve Şekil 4.8'e göre, eleman sayısı değiştirilerek gürültüsüz bir ortamda iş birliğine nasıl ulaşıldığı gösterilmiştir. Buna göre, sistemdeki elemanlara ait x bilgisi, denklem 4.1'de belirtilen iş birliği protokolüne göre ortak bir değere yakınsamıştır ve yeşil ile gösterilmiştir.



Şekil 4.6 : Çift entegratör sistemi için gürültüsüz ortam (eleman sayısı :30)

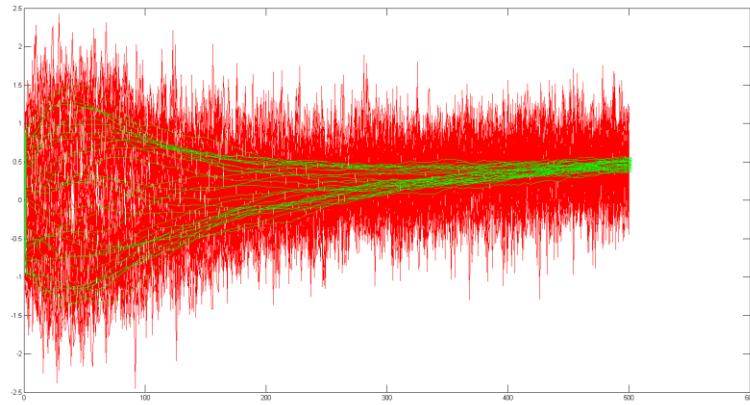


Şekil 4.7 : Çift entegratör sistemi için gürültüsüz ortam (eleman sayısı :20)

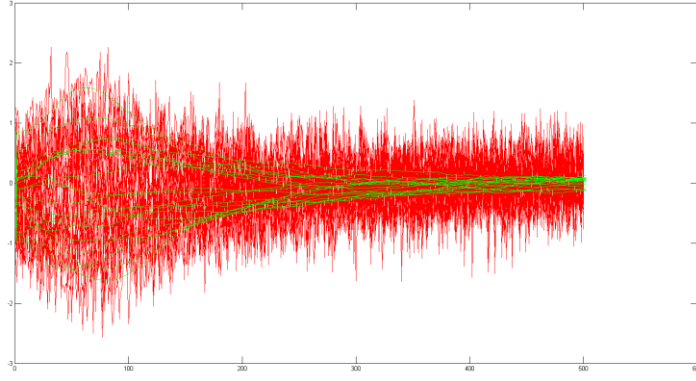


Şekil 4.8 : Çift entegratör sistemi için gürültüsüz ortam (eleman sayısı :10)

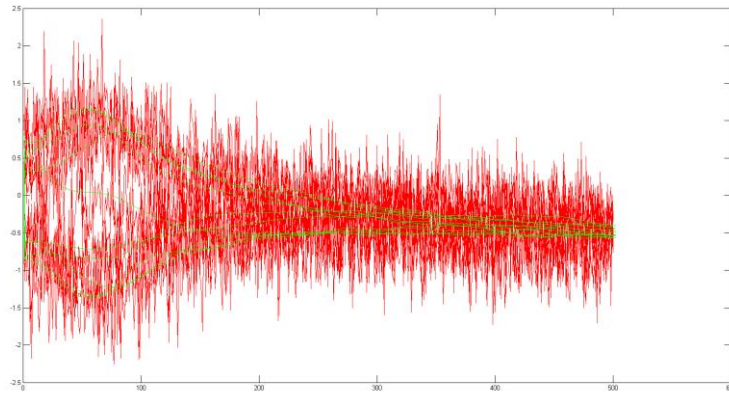
Şekil 4.9, Şekil 4.10 ve Şekil 4.11’de sisteme gürültü eklenmiştir. Sistemlerde bulunan eleman sayısı değiştirilerek gürültüye göre elemanların davranışı gözlenmiştir.



Şekil 4.9 : Çift entegratör sistemi için gürültülü ortam (eleman sayısı :30)

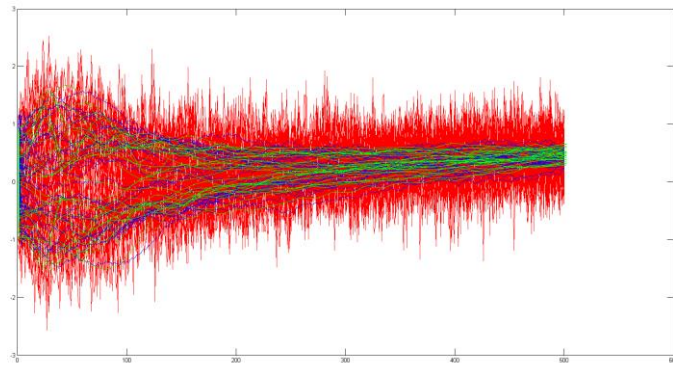


Şekil 4.10 : Çift entegratör sistemi için gürültülü ortam (eleman sayısı :20)

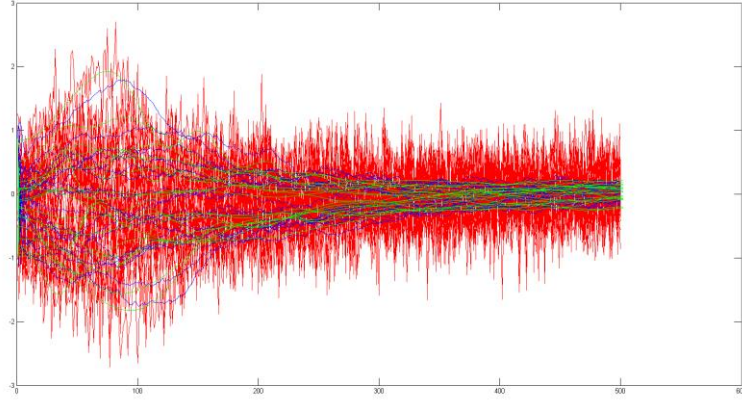


Şekil 4.11 : Çift entegratör sistemi için gürültülü ortam (eleman sayısı :10)

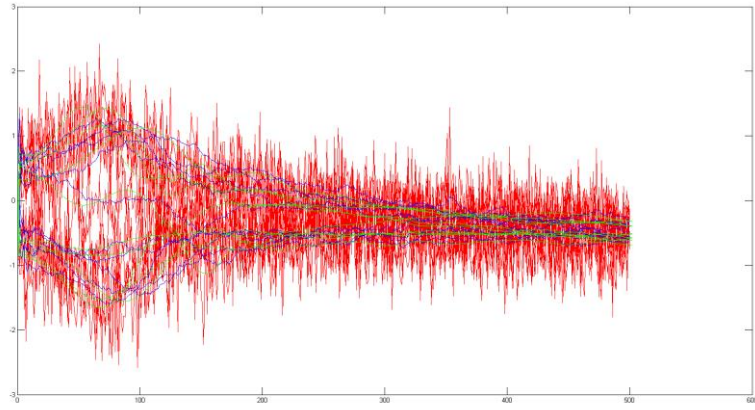
Buna göre yeşil ile gösterilen sistemdeki elemanların x bilgisi iken, kırmızı ile gösterilen ise sistemdeki ağ topolojisine göre her bir eleman için ölçülen değerleri ifade etmektedir.



Şekil 4.12 : Çift entegratör sistemi için Kalman filtresi uygulaması (eleman sayısı :30)



Şekil 4.13 : Çift entegratör sistemi için Kalman filtresi uygulaması (eleman sayısı :20)



Şekil 4.14 : Çift entegratör sistemi için Kalman filtresi uygulaması (eleman sayısı :10)

Sisteme ölçüm ve proses gürültüleri bozucu girdi olarak eklenmiştir. Buna göre elemanlar arasındaki bilgi alışverişi kırmızıyla gösterildiği gibi gürültülü hale getirilmiştir. Bu gürültülü ortamda sistemdeki elemanların yine de kontrol protokolüne ve dinamiğine bağlı olarak iş birliğine düzgün bir şekilde varabilmeleri için Kalman filtresi uygulanmıştır. Buna göre mavi ile gösterilen kırmızıyla alınan bilginin tahmin (estimate) edilmiş halidir.

4.2 Harmonik Osilator Sistemi

Bu bölümde çok etmenli bir sistem olan harmonik osilator sistemi ele alınacaktır. Söz konusu sistem belirli bir sayıda elemandan oluşmaktadır. Bu elemanlar arasında

bilgi alışverişi daha önce belirtilen ağ topolojisi üzerinden sağlanmakta ve anlaşılan bir değer üzerinde iş birliği sağlanmaktadır.

Mevcut sisteme ve sistemdeki elemanların bilgilerine gürültü eklenip, bozucu girdiler sisteme dahil edilmiştir. Söz konusu sistem lineer olduğu için temel Kalman filtresi denklemleri kullanılarak sistemin bu gürültülü ortamda bile sağlıklı bir şekilde iş birliğine varması sağlanmıştır.

Harmonik osilator sistemi, fiziksel olarak, herhangi bir kütleyle sahip bir cismin belirli bir kuvvetle hareket ettirilmesi sonrası, söz konusu cismin yer değiştirmesine bağlı bir kuvvetle geri çağırılması şeklinde düşünülebilir. İlerleyen bölümde söz konusu sisteme ait denklemler verilecektir. Söz konusu sistem harmonik osilator dinamiğine sahiptir ve elemanların bilgileri söz konusu dinamik çerçevesinde değişmektedir.

4.2.1 Sisteme ait parametre ve denklemler

Çok etmenli sisteme bir örnek olarak verilen harmonik osilator sistemi belirli parametrelerden oluşmaktadır. İlerleyen kısımlarda bu parametreler ve birbirleriyle olan ilişkileri belirten denklemlerden bahsedilecektir.

Harmonik osilator dinamiğine sahip bir sistem anlatılacaktır. Buna göre çok etmenli olan bu sistemde bulunan elemanlar harmonik osilator dinamiğine göre hareket edecek, yani sahip oldukları bilgiler bu söz konusu dinamiğe göre belirlenecektir.

Aşağıdaki denklemde harmonik osilator sistemine ait temel denklemler verilmiştir. Sistemdeki her bir elemanın davranışı bu denkleme göre şekillenecektir:

$$\dot{q}_i = p_i \quad i = 1, 2, \dots, n \quad (4.5)$$

$$\dot{p}_i = -w^2 q_i + u_i \quad i = 1, 2, \dots, n \quad (4.6)$$

$q_i \in R$, sistemdeki i 'inci elemanın konumu, $p_i \in R$, ise sistemdeki i 'inci elemanın hızı, $u_i \in R$, ise sistemdeki i 'inci elemana ait kontrol girdisi olarak düşünülebilir. w ise osilatorün frekansıdır.

Harmonik osilator sistemini oluşturan belirli sayıda eleman vardır. Bu elemanları tanımlamak için etmen (agent) ifadesi de kullanılmaktadır.

Sistemi oluşturan elemanlar belirli bilgilere sahiptir ve bu bilgiler bütün iterasyonlar için başlangıçta rastlantısal bir şekilde atanmıştır. Şekil 4.15'e de gösterildiği gibi başlangıçta her bir iterasyon için her bir elemanın "x" ve "xdot" bilgileri oluşturulmuştur. Şekil 4.15'de gösterilen “agents” eleman sayısını, “Niterations” ise iterasyon sayısını göstermektedir. Şekil 4.15'e göre belirlenen "x" ve "xdot" değişkenleri ölçüm parametrelerine atanmak üzere sırasıyla "xmeasured" ve "xdotmeasured" parametrelerine atanmıştır.

```

for t=1:Niterations
    for i=1:agents
        a=rand(1);
        if a >= 0.5
            b=1;
        else a < 0.5
            b=-1;
        end
        x(t,i)=1*b*rand(1);
        a=rand(1);
        if a >= 0.5
            b=1;
        else a < 0.5
            b=-1;
        end
        xdot(t,i)=1*b*rand(1);
    end
    xmeasured(t,:)=x(t,:);
    xdotmeasured(t,:)=xdot(t,:);
end

```

Şekil 4.15 : Sistemdeki elemanların bilgilerinin oluşturulması

```

for i=1:agents
    xlocation(i)=100*rand(1);
    ylocation(i)=100*rand(1);
end

for i=1:agents
    for j=1:agents
        A(i,j)=0;
    end
end

for i=1:agents
    for j=1:agents
        distance(i,j)=sqrt((xlocation(i)-xlocation(j))^2+(ylocation(i)-ylocation(j))^2);
        if distance(i,j) <= r_threshold
            A(i,j)=1;
        end
    end
end
end

```

Şekil 4.16 : Sistemdeki elemanların komşuluk ilişkilerinin oluşturulması

Daha önceki bölümlerde anlatıldığı gibi çok etmenli sistemlerde bulunan elemanlar birbirlerine belirli bir ağ topolojisine göre bağlıdır. Bu ağ topolojisinde bilgi transferi graf teorisinde de belirtildiği gibi komşuluk ilişkisine dayanır.

Şekil 4.16'da sistemi oluşturan elemanlar için belirli konum bilgileri rastlantısal olacak şekilde oluşturulmaktadır. Buna göre her bir eleman için x ve y koordinatlarındaki pozisyonlarını belirlemek üzere sırasıyla " $xlocation$ " ve " $ylocation$ " parametreleri oluşturulur.

Şekil 4.16'de belirtildiği gibi elemanlar arasındaki komşuluk aralarındaki mesafeye göre belirlenmektedir. Buna göre aralarındaki mesafeler hesaplanan elemanlar, bu mesafenin belirli bir değerin altında kalması durumunda komşudurlar ve ilgili " A " matrisinin ilgili hücrelerine 1 değeri atanır.

Söz konusu çok etmenli sistemde iş birliği protokolü katsayıları gradyant düşümü yöntemiyle ayarlanan bir adaptif PD ile sağlanır ve bu bütün elemanlar için bir kurallar bütünü olarak dışarıdan alınan bir girdi olarak kabul edilir.

Söz konusu sistemdeki kontrol girdisi her bir eleman için her bir iterasyonda güncellenecek şekilde oluşturulur ve Şekil 4.17'de gösterilen sum ile belirtilir.

Söz konusu çok etmenli sistem daha önce de belirtildiği gibi harmonik osilator dinamiğine sahiptir. Yani sistemi oluşturan elemanlar davranışlarını bu dinamiğe göre günceller.

```

for i=1:agents
    sum(t,i)=0;
    for j=1:agents
        if t == 1
            error(t,i,j)=xmeasured(t,i)-xmeasured(t,j);
            errordot(t,i,j)=xdotmeasured(t,i)-xdotmeasured(t,j);
        end
        if t > 1
            error(t,i,j)=xmeasured(t-1,i)-xmeasured(t-1,j);
            errordot(t,i,j)=xdotmeasured(t-1,i)-xdotmeasured(t-1,j);
        end
        term1=(error(t,i,j))*error(t,i,j);
        term2=errordot(t,i,j)*errordot(t,i,j);
        if t>1
            Kp(t,i,j)=Kp(t-1,i,j)-eta1*term1;
            Kd(t,i,j)=Kd(t-1,i,j)-eta2*term2;
        end
        sum(t,i)=sum(t,i)+Kp(t,i,j)*A(i,j)*error(t,i,j)+
            Kd(t,i,j)*A(i,j)*errordot(t,i,j);
    end
end

```

Şekil 4.17 : Sistemde bulunan elemanlar için kontrol girdilerinin oluşturulması

```

for i=1:agents
    fx(t,i)=-((omegan(i))^2)*x(t,i);
end

xdotdot(t,:)=fx(t,:)+sum(t,:)+w(t,:);
xdot(t+1,:)=xdot(t,:)+xdotdot(t,:)*dT;

x(t+2,:)=x(t+1,:)+xdot(t+1,:)*dT;
Xvector1(t,:)=x(t+2,:);
Xvector2(t,:)=xdot(t+1,:);
Xvector(t,:)=[Xvector1(t,:) Xvector2(t,:)];

z(t,:)=Xvector(t,:)+[v(t,:) v(t,:)];

for k=1:agents
    xmeasured(t,k)=z(t,k);
    xdotmeasured(t,k)=z(t,k+agents);
end

```

Şekil 4.18 : Sisteme gürütülerin eklenmesi

Daha önceki bölümlerde verilen denklem 4.2, 4.3 ve 4.4'den yararlanarak adaptif PD ve gradyant düşümü yardımıyla "*sum*" değişkeni ile belirtilen kontrol girdisi belirlenir.

Harmonik osilator dinamiği her bir eleman için " $fx(t,i)$ " yardımıyla oluşturulur. Daha sonra her bir elemanın matematiksel modelini belirleyen denklik, denklem 4.5 referans alınarak oluşturulur ve "*xdotdot*" değişkeni ile temsil edilir.

Sisteme proses ve ölçüm gürültüsü eklenerek sistemin davranışı gözlenir. Buna göre daha önceki bölümlerde de bahsedilen proses ve ölçüm gürültü kovaryansı parametrelerine göre (Q ve R) sisteme normal dağılıma sahip "*w*" ve "*v*" bozucu girdileri eklenir.

Sisteme bozucu girdilerin eklenmesiyle birlikte, sistemi oluşturan bilgilerin iş birliğine düzgün bir şekilde varabilmeleri için Kalman filtresi uygulanır ve buna bağlı tahminler geliştirilerek sistemi oluşturan elemanların bilgilerine geri besleme yapılır.

Sonuç olarak bir sonraki bölümde detaylı bir şekilde gösterilecek olan grafikler elde edilir.

4.2.2 Sisteme ait simülasyon sonuçları ve sonuçların yorumlanması

Çok etmenli sistemlere örnek olarak harmonik osilator sistemi incelenmiş ve ağ topolojisine sahip elemanların iş birliğine nasıl vardıkları üzerinde durulmuştur.

```

% xmeasured_estimate
if t==1
    xpri(t,:) = b.*xpost(t,:);
    ppri(:, :, t) = AA*ppost(:, :, t)*AA' + Q*(eye(agents));
end
if t>1
    xpri(t,:) = b.*xpost(t-1,:);
    ppri(:, :, t) = AA*ppost(:, :, t-1)*AA' + Q*(eye(agents));
    kk(:, :, t) = ppri(:, :, t) / (ppri(:, :, t) + R*eye(agents));
    xpost(t,:) = xpri(t,:) + (kk(:, :, t) *
        ((xmeasured(t,:) - xpri(t,:))'))';
    ppost(:, :, t) = (eye(agents) - kk(:, :, t)) * ppri(:, :, t);

% xdotmeasured_estimate
if t==1
    xpri2(t,:) = b.*xpost2(t,:);
    ppri2(:, :, t) = AA*ppost2(:, :, t)*AA' + Q*(eye(agents));
end
if t>1
    xpri2(t,:) = b.*xpost2(t-1,:);
    ppri2(:, :, t) = AA*ppost2(:, :, t-1)*AA' + Q*(eye(agents));
    kk2(:, :, t) = ppri2(:, :, t) / (ppri2(:, :, t) + R*eye(agents));
    xpost2(t,:) = xpri2(t,:) + (kk2(:, :, t) *
        ((xdotmeasured(t,:) - xpri2(t,:))'))';
    ppost2(:, :, t) = (eye(agents) - kk2(:, :, t)) * ppri2(:, :, t);

xmeasured(t,:) = xpost(t,:);
xdotmeasured(t,:) = xpost2(t,:);

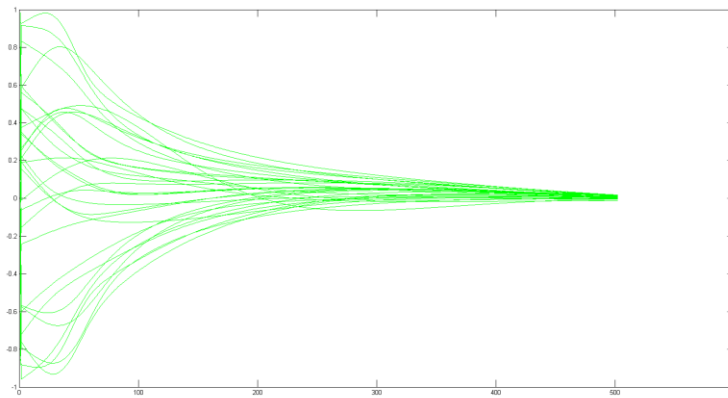
```

Şekil 4.19 : Kalman filtresi uygulanması

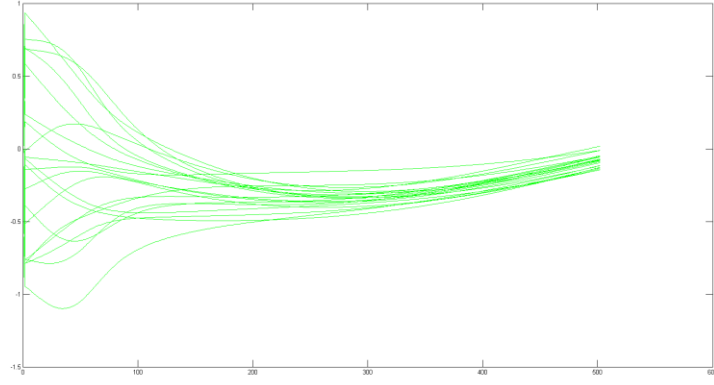
Aynı zamanda sisteme ait parametrelere gürültü eklenerek bozucu girdilerin sistem davranışına nasıl etki ettiği gösterilmiştir.

Kalman filtresi uygulanarak sistemin gösterdiği güncel davranış ilerleyen kısımlarda anlatılacaktır.

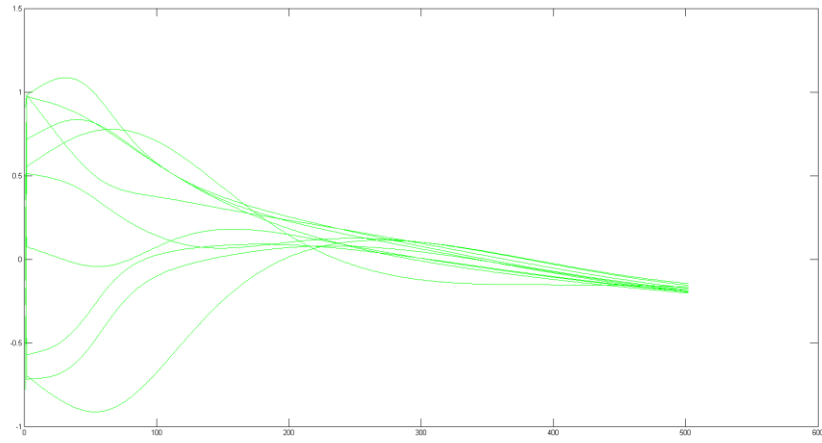
Şekil 4.20, Şekil 4.21 ve Şekil 4.22’de sistemde bulunan elemanların sayısı değiştirilerek elde edilen grafikler görülmektedir. Buna göre hem sistem dinamiği hem de kontrol girdisine göre sistemdeki elemanların sahip oldukları " x " değişkeninin değişimi yeşil ile gösterilmektedir.



Şekil 4.20 : Harmonik osilator sistemi için gürültüsüz ortam (eleman sayısı: 30)

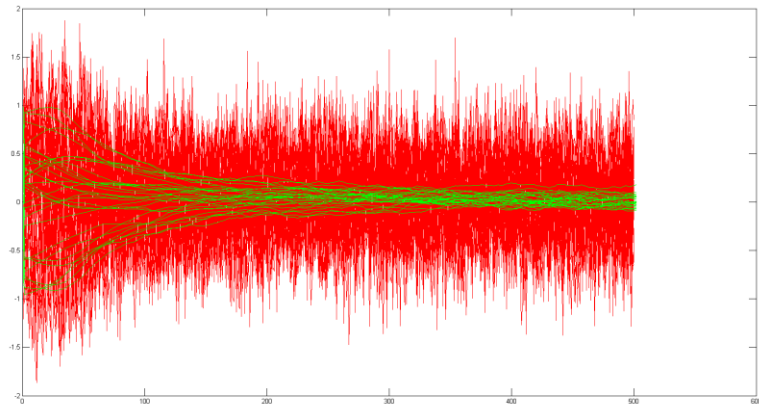


Şekil 4.21 : Harmonik osilator sistemi için gürültüsüz ortam (eleman sayısı: 20)

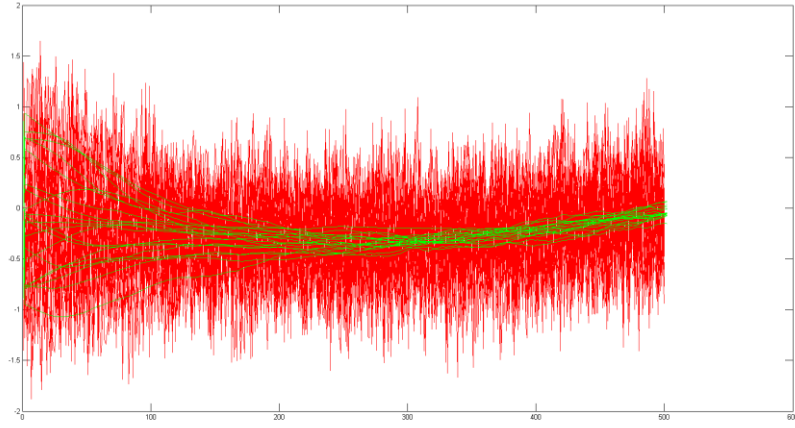


Şekil 4.22 : Harmonik osilator sistemi için gürültüsüz ortam (eleman sayısı: 10)

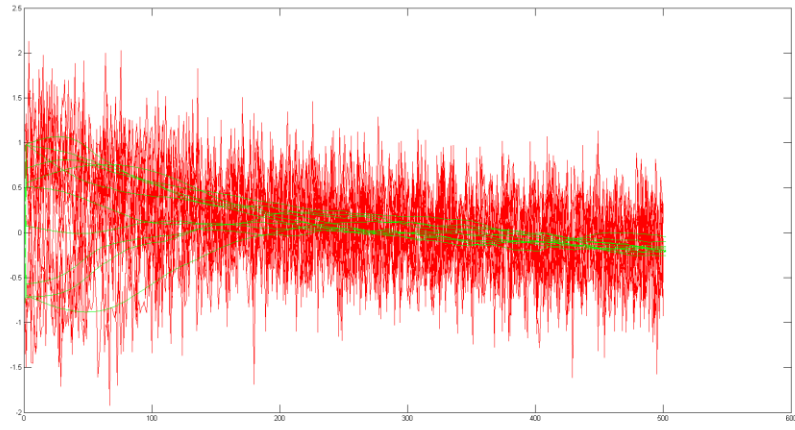
Şekil 4.23, Şekil 4.24 ve Şekil 4.25’de harmonik osilator dinamiğine sahip çok etmenli sisteme proses ve ölçüm gürültüsü eklenmiştir.



Şekil 4.23 : Harmonik osilator sistemi için gürültülü ortam (eleman sayısı: 30)



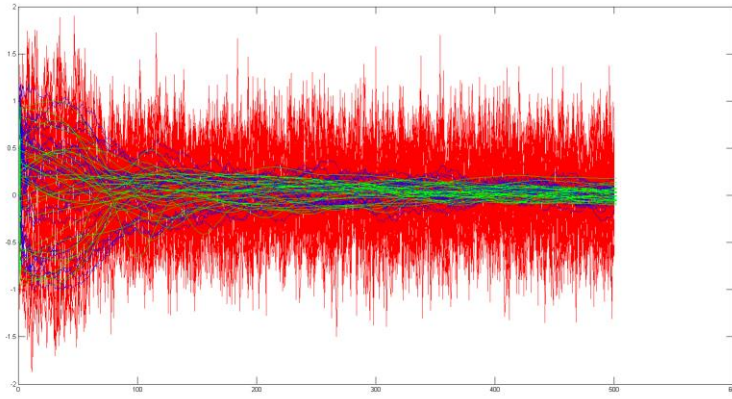
Şekil 4.24 : Harmonik osilator sistemi için gürültülü ortam (eleman sayısı: 20)



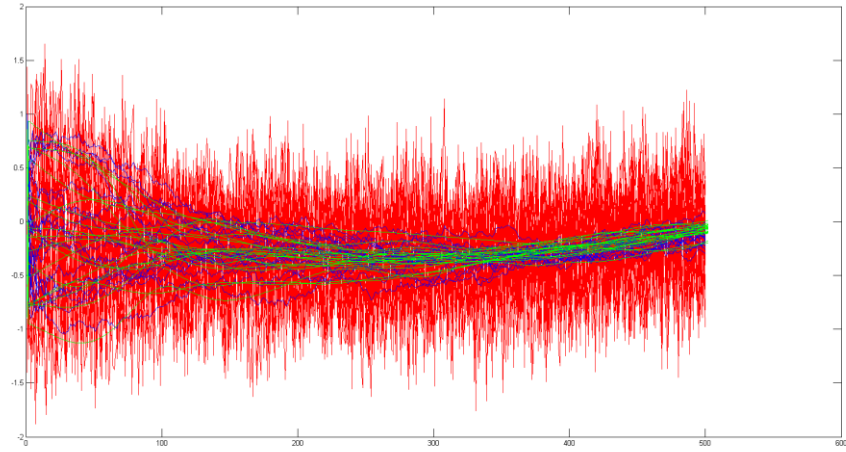
Şekil 4.25 : Harmonik osilator sistemi için gürültülü ortam (eleman sayısı: 10)

Her bir grafikte eleman sayısı değiştirilmiştir. Buna göre sistemdeki elemanlar, birbirlerinden bilgileri kırmızı ile gösterilen gürültü ile birlikte almaktadır. Elemanlardaki gerçek " x " bilgisi ise yeşil ile gösterilmektedir.

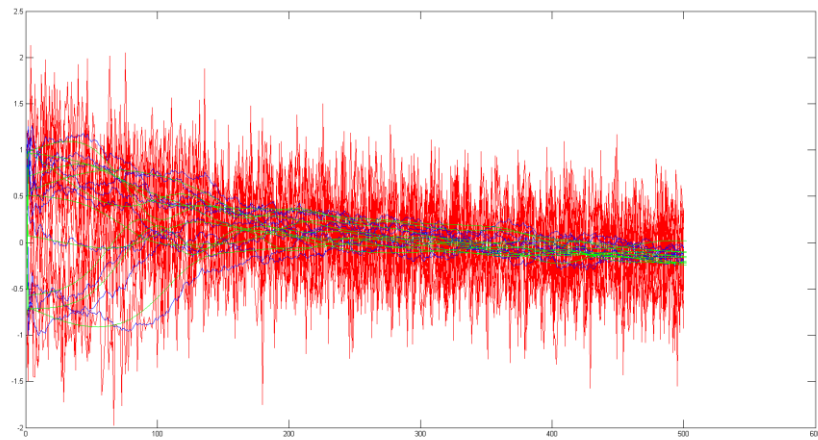
Şekil 4.26, Şekil 4.27 ve Şekil 4.28'e göre çok etmenli harmonik osilator sistemine hem ölçüm hem de proses gürültüsü eklenmiştir. Sistemdeki elemanlar birbirlerinden aldıkları gürültülü bilgilere rağmen uygulanan Kalman filtresi sayesinde daha düzgün bir biçimde iş birliğine ulaşmıştır. Buna göre elemanların aldıkları gürültülü bilgiler kırmızıyla, bu gürültülü ortamda Kalman filtresi sayesinde geliştirdikleri tahmin (esitmate) değerleri ise mavi ile gösterilmiştir. Sistemdeki elemanlar bulundurdıkları bilgileri ürettikleri tahmin değerleriyle birlikte güncellemiştir. Grafikte yeşil ile gösterilen " x " bilgileri sistemdeki elemanların güncel bilgileridir.



Şekil 4.26 : Harmonik osilator için Kalman filtresi uygulaması (eleman sayısı:30)



Şekil 4.27 : Harmonik osilator için Kalman filtresi uygulaması (eleman sayısı:20)



Şekil 4.28 : Harmonik osilator için Kalman filtresi uygulaması (eleman sayısı:10)

4.3 Sinüzoidal Sistem

Bu bölümde çok etmenli bir sistem olan sinüzoidal sistem ele alınacaktır. Söz konusu sistem belirli sayıda elemandan oluşmaktadır. Bu elemanlar arasında bilgi alışverişi daha önce belirtilen bir ağ topolojisi üzerinden sağlanmakta ve anlaşılan bir değer üzerinde iş birliği sağlanmaktadır.

Sinüzoidal sistem, harmonik sistem ile benzer karakteristiktir. Yani sinüzoidal sistemi, hareketli bir cismin belirli bir denge konumundan, diğer denge konumuna doğru olan uzaklığıyla ters orantılı ama ters yönlü bir ivmeyle yaptığı düzenli titreşimler olarak düşünebiliriz. Bu hareket zamana bağlı bir sinüs eğrisiyle gösterilebilir ve düzenli olarak tekrarlanır.

Yapılan çalışmada sinüzoidal sistem dinamiğine sahip çok etmenli sisteme ait elemanların birbirleriyle haberleşmelerine ve mevcut sisteme gürültü eklenecektir. İncelenen söz konusu sistem lineer olmadığı için bu gibi durumlarda kullanılan Kalman filtresinin bir uygulaması olan, genişletilmiş Kalman filtresi (extended kalman filter) kullanılacaktır.

4.3.1 Sisteme ait parametre ve denklemler

Çok etmenli bir sisteme örnek olarak verilen sinüzoidal sistem incelenecektir. Sisteme ait denklem aşağıdaki gibidir:

$$f(x) = b * amp * (\sin(x) - \frac{3.14}{phase}) \quad (4.7)$$

Söz konusu çok etmenli sinüzoidal sisteme ait elemanlar denklem 4.4'e göre bilgilerini güncellemektedir. Verilen denklemdeki "*amp*" parametresi şiddeti, "*phase*" ise faz farkını göstermektedir.

```
for t=1:Niterations
    for i=1:agents
        x(t,i)=(3.14/180)*100*rand(1);
        xdot(t,i)=xdot_coeff*rand(1);
    end
end
for t=1:Niterations
    xmeasured(t,:)=x(t,:);
    xdotmeasured(t,:)=xdot(t,:);
end
```

Şekil 4.29 : Sistemdeki elemanların bilgilerinin oluşturulması

Şekil 4.29’da gösterildiği gibi söz konusu çok etmenli sistemde bulunan elemanlar için rastlantısal olacak şekilde " x " ve " $\dot{x}$ " değerleri tanımlanmıştır. Bu değerler, sistemdeki elemanların sahip oldukları bilgilerdir. Söz konusu " x " ve " $\dot{x}$ " değerleri sırasıyla " $x_{measured}$ " ve " $\dot{x}_{measured}$ " parametrelerine atanarak ölçüm değerleri oluşturulmuştur.

```

for i=1:agents
    xlocation(i)=100*rand(1);
    ylocation(i)=100*rand(1);
end
for i=1:agents
    for j=1:agents
        A(i,j)=0;
    end
end
for i=1:agents
    for j=1:agents;
        distance(i,j)=sqrt((xlocation(i)-xlocation(j))^2+
            (ylocation(i)-ylocation(j))^2);
        if distance(i,j) <= r_threshold;
            A(i,j)=1;
        end
    end
end
for i=1:agents;
    A(i,agents)=1;
end

```

Şekil 4.30 : Sistemdeki elemanların komşuluk ilişkilerinin oluşturulması

Değerler başlangıçta her bir iterasyonu kapsayacak şekilde oluşturulsa da sistem dinamiği ve kontrol girdilerine bağlı olarak güncellenecektir.

```

for i=1:agents
    sum(t,i)=0;
    for j=1:agents
        if t == 1
            error(t,i,j)=xmeasured(t,j)-xmeasured(t,i);
            errordot(t,i,j)=xdotmeasured(t,j)-xdotmeasured(t,i);
        end
        if t > 1
            error(t,i,j)=xmeasured(t-1,j)-xmeasured(t-1,i);
            errordot(t,i,j)=xdotmeasured(t-1,j)-xdotmeasured(t-1,i);
        end
        term1=error(t,i,j)*error(t,i,j);
        term2=errordot(t,i,j)*errordot(t,i,j);
        if t>1
            Kp(t,i,j)=Kp(t-1,i,j)-(eta1(i)/N(i))*term1;
            Kd(t,i,j)=Kd(t-1,i,j)-(eta2(i)/N(i))*term2;
        end
        if (t*dT) <= delay
            sum(t,i)=sum(t,i)+0;
        end
        if (t*dT) > delay
            sum(t,i)=sum(t,i)+Kp(t,i,j)*A(i,j)*error(t,i,j)+
                Kd(t,i,j)*A(i,j)*errordot(t,i,j);
        end
    end
end

```

4.31 : Sistemde bulunan elemanlar için kontrol girdisinin oluşturulması

Şekil 4.30’a göre sistemde bulunan elemanlar için x ve y koordinatlarında olmak üzere konum bilgileri rastlantısal olarak oluşturulmaktadır. Daha sonrasında ise,

birbirlerine olan mesafesi belirli bir değerin altında olan elemanlar komşu olarak belirlenmektedir ve A matrisi buna göre oluşturulmaktadır.

Daha önceki bölümlerde verilen denklem 4.2, 4.3 ve 4.4'den yararlanarak adaptif PD ve gradyant düşümü yöntemi yardımıyla " sum " değişkeni ile gösterilen kontrol girdisi belirlenir. Buna göre her bir elemana her bir iterasyon sırasında belirlenen kontrol girdisi etki edecektir ve bu sayede iş birliği elde edilecektir.

```
for i=1:agents
    fx(t,:)=b(i)*amplitude(i)*sin(x(t,i)-3.14/phase(i))+w(t,:);
end
    xdotdot(t,:)=fx(t,:)-sum(t,:);
    xdot(t+1,:)=xdot(t,:)+xdotdot(t,:)*dT;
    x(t+2,:)=x(t+1,:)+xdot(t+1,:)*dT;
    Xvector1(t,:)=x(t+2,:);
    Xvector2(t,:)=xdot(t+1,:);
    Xvector(t,:)= [Xvector1(t,:) Xvector2(t,:)];
    z(t,:)=Xvector(t,:)+[v(t,:) v(t,:)];
for k=1:agents
    xmeasured(t,k)=z(t,k);
    xdotmeasured(t,k)=z(t,k+agents);
end
```

Şekil 4.32 : Sisteme gürültelerin eklenmesi

Şekil 4.32'de de gösterildiği gibi söz konusu sinüzoidal sistemin dinamiği " $fx(t,i)$ " ile belirlenmiştir. " $xdotdot$ " parametresi ise sistemin dinamiği ile kontrol girdisinin biraraya getirildiği matematiksel modelini temsil etmektedir. Daha sonrasında sisteme proses ve ölçüm gürültüleri eklenmiştir. Eklenen bu değerler her bir iterasyon da kullanılmak üzere " $xmeasured$ " ve " $xdotmeasured$ " değerlerine atanmıştır.

```
if t==1
    xpri(t,:)=bb.*xpost(t,:);
    ppri(:, :, t)=AA*ppost(:, :, t)*AA'+Q*(eye(agents));
end
if t>1
    xpri(t,:)=bb.*xpost(t-1,:);
    ppri(:, :, t)=AA*ppost(:, :, t-1)*AA'+Q*(eye(agents));
    kk(:, :, t)=ppri(:, :, t)/(ppri(:, :, t)+R*eye(agents));
    xpost(t,:)=xpri(t,:)+(kk(:, :, t)*(xmeasured(t,:)-xpri(t,:)))';
    ppost(:, :, t)=(eye(agents)-kk(:, :, t))*ppri(:, :, t);
end
if t==1
    xpri2(t,:)=bb.*xpost2(t,:);
    ppri2(:, :, t)=AA*ppost2(:, :, t)*AA'+Q*(eye(agents));
end
if t>1
    xpri2(t,:)=bb.*xpost2(t-1,:);
    ppri2(:, :, t)=AA*ppost2(:, :, t-1)*AA'+Q*(eye(agents));
    kk2(:, :, t)=ppri2(:, :, t)/(ppri2(:, :, t)+R*eye(agents));
    xpost2(t,:)=xpri2(t,:)+(kk2(:, :, t)*(xdotmeasured(t,:)-xpri2(t,:)))';
    ppost2(:, :, t)=(eye(agents)-kk2(:, :, t))*ppri2(:, :, t);
end
```

Şekil 4.33 : Genişletilmiş Kalman filtresinin uygulanması

Şekil 4.33'e göre söz konusu sinüzoidal sisteme proses ve ölçüm gürültüsü eklendikten sonra bu gürültülü ortamda sistem elemanları sahip oldukları bilgilerle

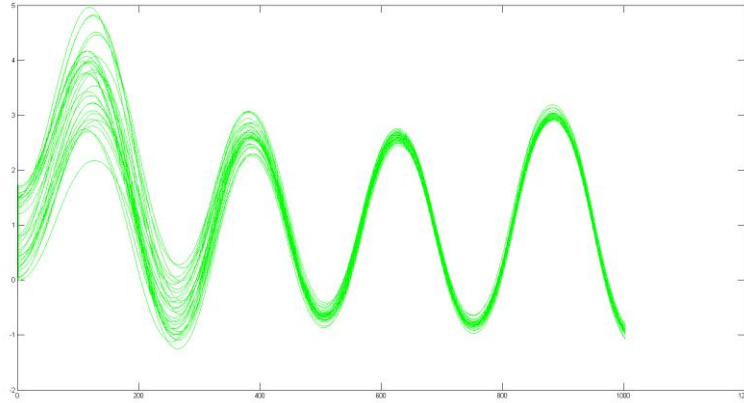
ilgili genişletilmiş Kalman filtresi sayesinde tahminde bulunarak mevcut bilgilerini günceller. Genişletilmiş Kalman filtresi önceki bölümlerde verilen denklemlere göre uygulanmıştır.

4.3.2 Sisteme ait simülasyon sonuçları ve sonuçların yorumlanması

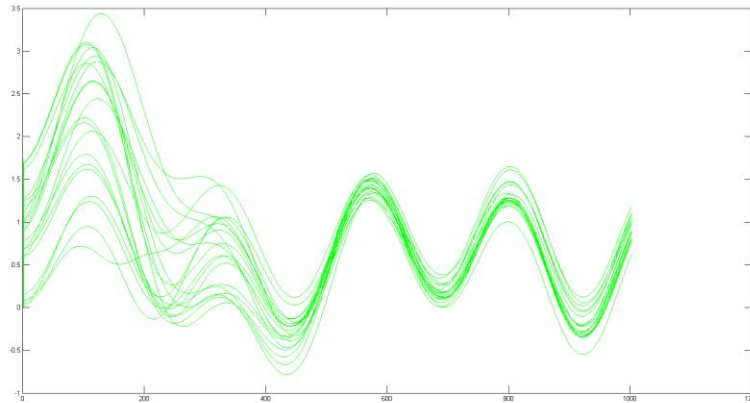
Çok etmenli sistemlere örnek olarak sinüzoidal bir sistem ele alınmıştır. Söz konusu sisteme gürültü eklenerek sistemin cevabı incelenmiştir.

Aynı zamanda birbirleriyle gürültülü bilgi alışverişi yapan elemanların sahip oldukları bilgilerdeki değişim gözlenmiştir. Daha sonrasında ise sisteme genişletilmiş Kalman filtresi uygulanarak sistemdeki elemanların davranışı gözlenmiştir.

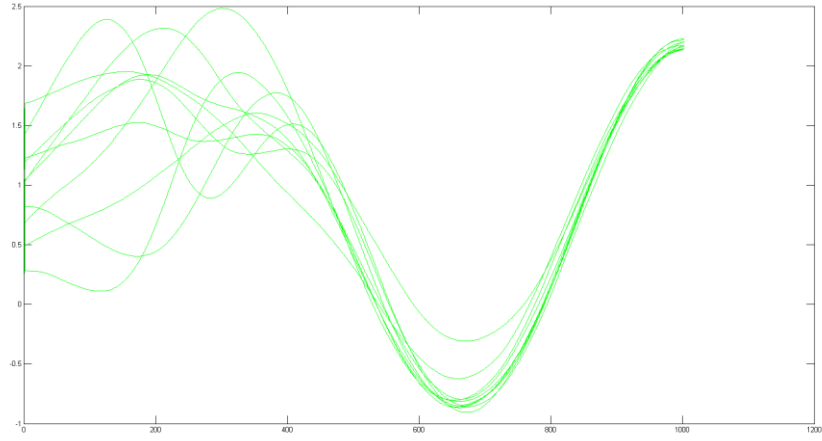
Şekil 4.34, Şekil 4.35 ve Şekil 4.36’da söz konusu sisteme ait elemanların x bilgilerinin sinüzoidal sistem dinamiği ve iş birliği kontrol protokolü girdileriyle birlikte nasıl iş birliğine vardıkları gözlenmektedir.



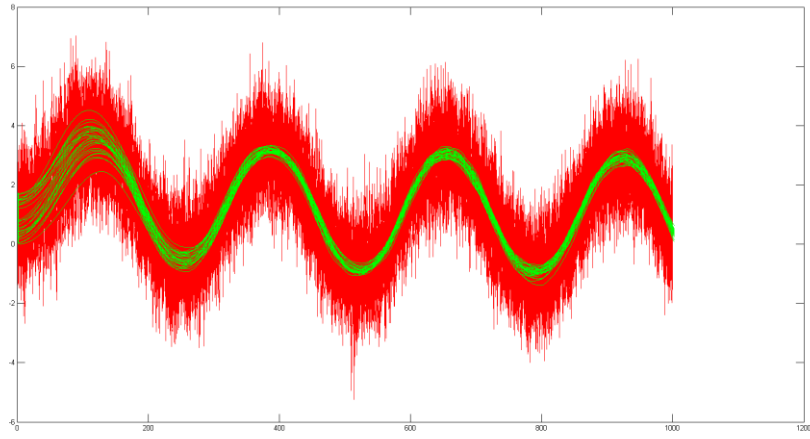
Şekil 4.34 : Sinüzoidal sistem için gürültüsüz ortam (eleman sayısı: 30)



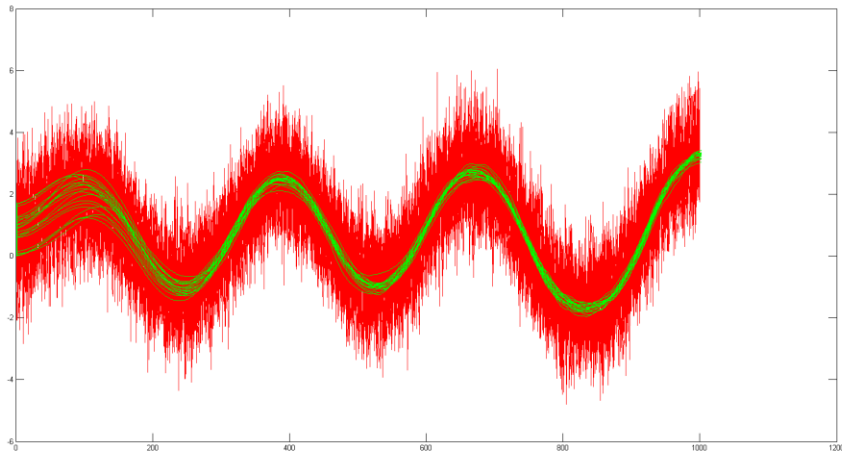
Şekil 4.35 : Sinüzoidal sistem için gürültüsüz ortam (eleman sayısı: 30)



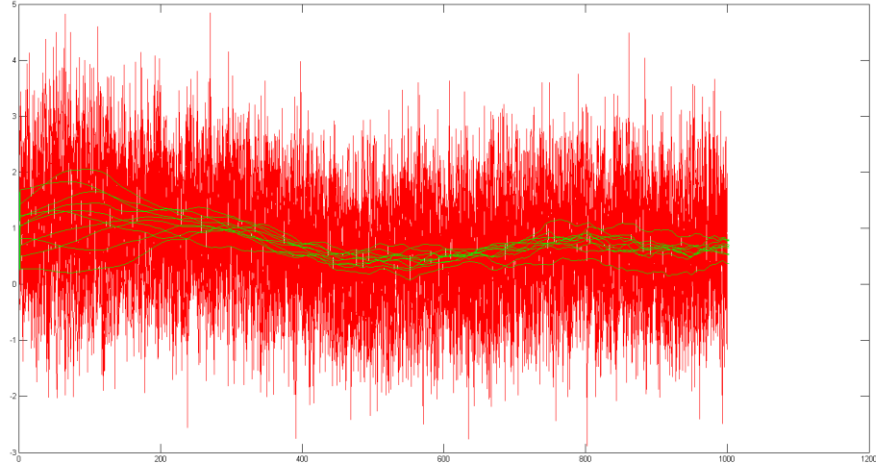
Şekil 4.36 : Sinüzoidal sistem için gürültüsüz ortam (eleman sayısı: 10)



Şekil 4.37 : Sinüzoidal sistem için gürültülü ortam (eleman sayısı: 30)



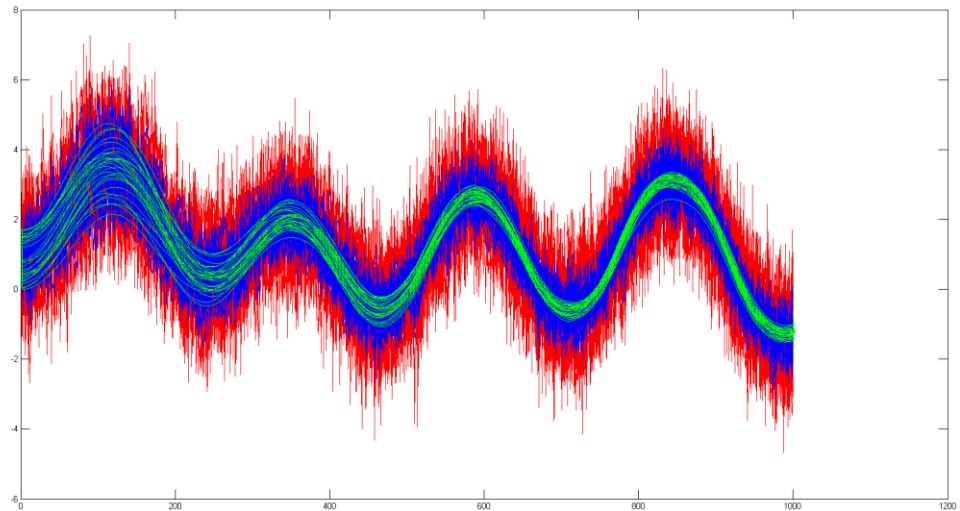
Şekil 4.38 : Sinüzoidal sistem için gürültülü ortam (eleman sayısı: 20)



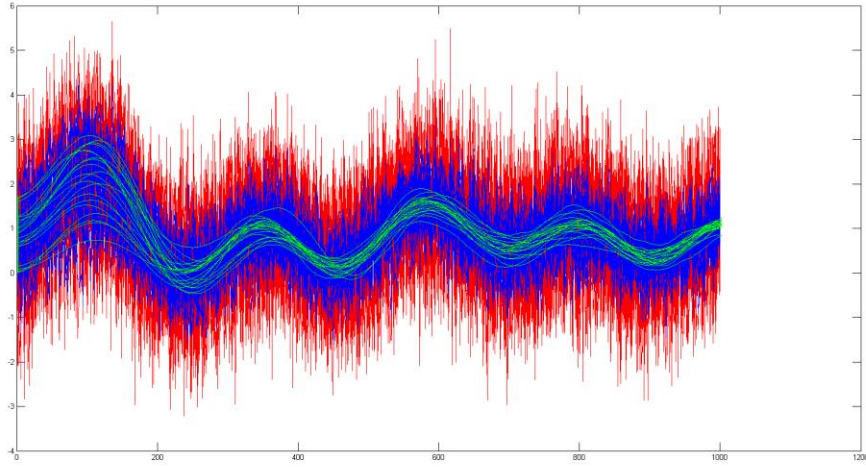
Şekil 4.39 : Sinüzoidal sistem için gürültülü ortam (eleman sayısı: 10)

Şekil 4.37, Şekil 4.38 ve Şekil 4.39’da sisteme gürültü eklenerek farklı eleman sayılarına sahip sistemin her bir durumda nasıl davrandığı gözlenmiştir. Buna göre kırmızı ile gösterilenler sistemdeki elemanların gürültülü bilgilerdir. Yeşil ile ise gürültüsüz "x" bilgilerinin davranışları gösterilmektedir.

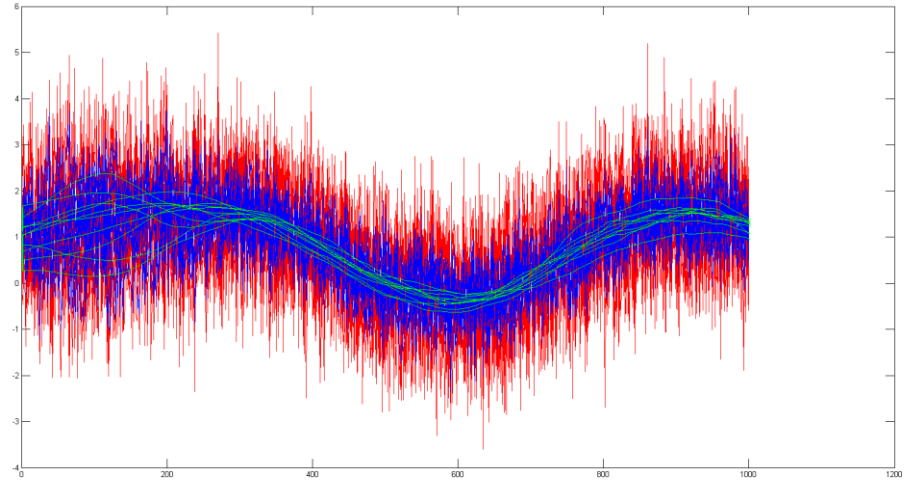
Şekil 4.40, Şekil 4.41 ve Şekil 4.42’da ise sisteme gürültü eklendikten sonra Kalman filtresi uygulanmıştır ve sonuçlar aynı sistem ve dinamikte farklı sayıda eleman için incelenmiştir. Buna göre, mavi ile gösterilen sistemdeki her bir elemanın ürettiği tahmin (estimate) değeridir.



Şekil 4.40 : Sinüzoidal sistem için Kalman filtresi uygulaması (eleman sayısı :30)



Şekil 4.41 : Sinüzoidal sistem için Kalman filtresi uygulaması (eleman sayısı :20)



Şekil 4.42 : Sinüzoidal sistem için Kalman filtresi uygulaması (eleman sayısı :10)

5. SONUÇLAR

Bu bölümde, yapılan çalışma özetlenerek ileride yapılabilecek olan çalışmalar hakkında bilgiler verilecektir.

Yapılan tez çalışmasında çok etmenli sistemler incelenmiştir. Buna göre çok etmenli sistemler birden fazla elemanın biraraya gelerek belirli amaç ve görev doğrultusunda iş birliği içerisinde davrandığı sistemlerdir. Bu gibi sistemler günümüzde otonom ve robotik sistemlerin artmasıyla birlikte daha yaygın bir biçimde kullanılmaktadır.

İncelenen çok etmenli sistemlerdeki elemanlar için iş birliği koşulları araştırılmıştır. Bir arı sürüsü gibi sisteme yayılan bu elemanlar görünmeyen bir ağ üzerinden belirli bir ağ topolojisi yardımıyla birbirleriyle haberleşebilmektedir.

Çalışmada görüldüğü gibi bu ağ yapısındaki elemanlar belirli komşuluk ilişkileri ve iş birliği protokolleri üzerinden durumlarını diğer elemanlara göre güncellemekte ve bir süre sonra tek bir eleman gibi hareket etmeye başlamakta veya bu duruma yakınsamaktadır.

Gerçek dünyadaki tekil veya çok etmenli sistemler matematiksel modele uygun olarak hareket etmeye çalışmada söz konusu sistemlerin dinamiğine veya davranışına etki edebilen bozucu girdiler (gürültüler) söz konusudur.

Yapılan çalışmada aynı zamanda sisteme etki edebilecek bozucu girdiler için teorik altyapı oluşturulmuştur. Buna göre yapılacak çalışmalara ön ayak olması için olasılık ve normal dağılımlar hakkında bilgi verilerek hem bu bozucu girdiler hem de sistemin ve sistemdeki elemanların bu bozucu girdilere karşı tepkisi hakkında fikir vermeye çalışılmıştır.

Çok etmenli sistemlere ve sistemde birbirleriyle haberleşen elemanlar arasındaki bilgi alışverişine yapay olarak eklenen bu gürültü girdileri, gürültü genliği ölçüsünde sistemin iş birliğine varabilmesini de olumsuz olarak etkilediği yapılan çalışmalarda görülmüştür.

Gerçek dünyada karşılaşılan bu gürültülü ortamlara rağmen çok etmenli sistemlerdeki elemanların düzgün bir şekilde iletişime geçip iş birliğine ulaşmalarını sağlayabilmek adına etkili bir tahmin algoritması olan Kalman filtresinden yararlanılmıştır. Yapılan tez çalışmasında teorisi anlatılan söz konusu tahmin algoritmasının çeşitli sistemlerde doğrulaması yapıp, iletişim topolojileri üzerindeki olumlu etkileri gözlemlenmiştir.

Yapılan çalışma sayesinde, birbirleriyle haberleşen elemanlardan oluşan çok etmenli sistemlerin çeşitli bozucu girdi veya olaylara rağmen Kalman filtresi sayesinde tahmin geliştirip iş birliğine sağlıklı bir şekilde ulaşabileceği görülmüştür.

Her bir sistem için gerçekleştirilen Kalman filtresi uygulamaları sonucu belirlenen tahmin değerlerinin iterasyon boyunca gerçek değerlere olan yakınsamasını gözlemlemek adına L_2 normundan yararlanılmıştır. L_2 normu için ilgili denklem aşağıdaki gibidir [32]:

$$|x| = \sqrt{\sum_{k=1}^n |x_k|^2} \quad (5.1)$$

Denklem 5.1’de verilen x , $x = [x_1 \ x_2 \ \dots \ x_n]$ şeklinde olmak üzere vektör formundadır.

x_{hata1} değeri Kalman filtresi uygulanmayan gürültülü bir ortamda denklem 5.2’ye göre hesaplanır.

$$x_{hata1} = x_{ölçüm} - x_{gerçek} \quad (5.2)$$

x_{hata2} değeri Kalman filtresi uygulanan gürültülü bir ortamda denklem 5.3 göre hesaplanır.

$$x_{hata2} = x_{tahmin} - x_{gerçek} \quad (5.3)$$

4. bölümde bahsedilen; " x " değişkeni, denklem 5.2 ve denklem 5.3’de gösterilen $x_{gerçek}$ değerini temsil eder. 4. bölümde bahsedilen " x_{post} " değişkeni ise denklem 5.3’deki x_{tahmin} değerini temsil ederken, yine 4. bölümde bahsedilen " $x_{measured}$ " değişkeni ise denklem 5.2’de gösterilen $x_{ölçüm}$ ile temsil edilir.

Yapılan çalışmada görüldüğü gibi Kalman filtresi uygulanarak elde edilen tahminlerdeki hatalar, her bir sistem için iterasyon sayısı arttıkça azalmaktadır. Sistem zamanla geçtikçe belirli bir değere kadar $x_{gerçek}$ değerine yakınsamaya başlamaktadır.

Çizelge 5.1, Çizelge 5.2 ve Çizelge 5.3’de elde edilen değerler görülmektedir. Söz konusu çizelgelerde belirtilen L_2 (a) gürültülü ortamda Kalman filtresi uygulanmadan elde edilmiştir. L_2 (b) ise gürültülü ortamda Kalman filtresi uygulanarak elde edilen tahmin değerlerine göre belirlenmiştir.

Çizelge 5.1 : Çift entegratör sistemi için L_2 norm.

İterasyon Aralığı	$x_{gerçek}$	$x_{ölçüm}$	x_{tahmin}	L_2 (a)	L_2 (b)
0-50	886,8	1059,9	821,2	602,4	302,4
51-100	1093,9	1215,8	962,5	613,5	437,3
101-150	579,2	829,1	780,0	618,5	590,8
151-200	518,6	795,8	543,7	606,0	470,8
201-250	381,2	727,8	401,9	600,8	271,8
251-300	289,9	668,2	299,1	595,3	281,4
301-350	273,5	661,4	245,6	602,1	193,8
351-400	219,0	636,1	207,8	608,7	151,8
401-450	204,6	639,5	193,3	600,5	123,1
451-500	203,6	632,3	190,8	604,1	118,2
Toplam	4650,3	7865,9	4645,8	6051,9	2941,3

Çizelge 5.2 : Harmonik Osilatör sistemi için L_2 norm.

İterasyon Aralığı	$x_{gerçek}$	$x_{ölçüm}$	x_{tahmin}	L_2 (a)	L_2 (b)
0-50	494,5	585,5	472,7	398,3	140,0
51-100	319,7	494,7	424,8	392,6	211,2
101-150	286,1	506,2	252,9	417,5	274,7
151-200	275,9	497,4	209,2	388,2	221,5
201-250	173,7	436,2	153,4	396,9	179,0
251-300	187,0	428,2	132,4	387,9	142,3
301-350	120,7	417,8	113,6	395,5	122,3
351-400	108,0	413,7	97,9	394,9	87,3
401-450	95,2	402,0	72,9	387,1	114,7
451-500	99,5	411,4	53,2	403,2	84,8
Toplam	2160,3	4606,3	1983,1	3962,2	1577,8

İleride yapılacak çalışmalarda, sistemlerde yaşanan çeşitli haberleşme kesintileri, doğal afetler vb. yaşanan olumsuzluklarda bile sistemlerdeki elemanların öğrenme kabiliyetlerinin artırılması sayesinde kolayca tahmin geliştirip, sistemi söz konusu olumsuzluklar yaşanmamış gibi bozmadan çalıştırabilmesi hedeflenmektedir.

Çizelge 5.3 : Sinozaidal sistem için L_2 norm.

İterasyon Aralığı	$X_{\text{gerçek}}$	$X_{\text{ölçüm}}$	X_{tahmin}	$L_2(a)$	$L_2(b)$
0-50	414,5	562,3	545,1	397,8	107,7
51-100	282,9	501,7	235,9	418,0	240,1
101-150	1024,5	1070,2	743,8	399,2	280,9
151-200	1282,2	1259,1	1204,6	407,3	111,4
201-250	800,1	833,2	975,3	405,5	193,2
251-300	437,0	628,6	407,4	415,5	340,5
301-350	947,8	983,6	729,0	395,2	234,6
351-400	1246,4	1268,1	1133,3	387,7	162,7
401-450	1975,9	1990,7	1670,3	380,5	306,7
451-500	3164,0	3193,0	3212,2	379,3	365,3
Toplam	11575,3	12290,5	10765,9	3986,0	2343,0

KAYNAKLAR

- [1] **R.O. Saber, J. A. Fax ve R. M. Murray** (2007). Algorithms that provide rapid agreement and teamwork between all participants allow effective task performance by self-organizing networked systems. *Proceeding of the IEEE*. (pp. 215-231)
- [2] **R.C. Gomez ve N. Olgac** (2011). Exhaustive stability analysis in a consensus system with time delay and irregular topologies. *International Journal of Control*. (pp. 747-757)
- [3] **W. Ren ve R. W. Beard** (2008). *Distributed Consensus in Multi-vehicle Cooperative Control, Theory and Applications*. (pp. 3-22)
- [4] **S. Russel ve P. Norvig** (1995) Artificial intelligence-A modern approach, *Prentice Hall*
- [5] **P. Maess** (1995) Artificial life meets entertainment: Life like autonomous agents, *Communications of the ACM*, vol. 38, no.11, pp (108-114)
- [6] **P. Stone ve M. Veloso** (2000) Multiagent systems: A surevey from a machine learning perspective, *Autonomous Robots*, vol. 8, no.3, pp (345-383)
- [7] **T. Zhang ve H. Yu** (2010). Average Consensus in Networks of Multi-Agent with Multiple Time-Varying Delays. *Int. J. Communications, Network and System Sciences* (pp. 196-203)
- [8] **A. Periaipoilan** (2013, December). *Study of Consensus Algorithm and Decentralized Formation Control of Autonomous Vehicles Applied to the Context of Unmanned Disaster Response System*. (Yüksek Lisans tezi), California State University, Northridge, USA
- [9] **J. A. Bondy ve U. S. R. Murty** (2008) Graph Theory, *New York: Springer Verlag*.
- [10] **C. Martensson ve L. Sjoval** (2011, May). *Consensus Algorithms, Flocking and Swarms* (Lisans tezi), Department of Mathematics, Optimization and Systems Theory Royal Institute of Thechnology Stockholm, Sweden
- [11] **J. Wu** (2013, May). *Consensus in Multi-Agent Systems and Bilateral Teleoperation with Communication Constraints* (Doktora tezi), Department of Mechanical Engineering, University of Victoria
- [12] **A. Pandey** (2012). *Consensusability of discrete-time multi-agent systems* (Yüksek Lisans tezi), Louisiana State University and Agricultural and Mechanical College, USA

- [13] **W. Qiang ve W. Yuzhen** (2012, July). Consensus of a Class of Multi-Agent Systems Based on Directed Graph: Fixed and Switching Topology, *Proceedings of the 31st Chinese Control Conference*
- [14] **Y. Cao ve W. Ren** (2010) Multi-Agent Consensus Using Both Current and Outdated States with Undirected Interaction, *J. Intell. Robotics Syst.* (pp. 95-106)
- [15] **J. Hu, Y. Li, T. Yong, J. Cao, J. Yu ve W. Mao** (2014, August) Distributed Cooperative Regulation for Multiagent Systems and Its Applications to Power Systems: A Survey, *The Scientific World Journal.* (pp. 1-17)
- [16] **W. Ren, R. W. Beard ve E. M. Atkins** (2005). A Survey of Consensus Problems in Multi-agent Coordination. *American Control Conference.* (pp. 1859-1864)
- [17] **Z. Han, Z. Lin, M. Fu ve Z. Chien** (2015, April). Distributed Coordination in Multi-agent Systems: A Graph Laplacian Perspective. *Frontiers of Information Technology & Electronic Engineering*
- [18] **C. L. Liu ve F. Liu** (2010). Consensus Problems of Second-order Dynamic Agents with Heterogeneous Input and Communication Delays. *Int. J. of Computers, Communications & Control*
- [19] **L. Liu** (2016, June). *Cooperative Control of Nonlinear Multi-Agent Systems* (Doktora tezi), Graduate Program in Earth and Space Science York University, Toronto, Ontario, USA
- [20] **J. Xie ve C.C. Liu** (2017, June). Multi-agent systems and their applications, *Journal of International Council on Electrical Engineering* (pp. 188-197)
- [21] **R.O. Saber** (2006, March). Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory. *IEEE Transactions on Automatic Control* (pp. 401-419)
- [22] **V. O. Karasev ve V. A. Sukhanov** (2016, October) Product Lifecycle Management using multi-agent systems models, *XIIth International Symposium «Intelligent Systems», INTELS'16, 5-7 October 2016, Moscow, Russia*
- [23] **F. Munoz, E. S. E. Quesada, H. M. La, S. Salazar, S. Commuri ve L. R. G. Carillo** (2006, March). Adaptive consensus algorithms for real-time operation of multi-agent systems affected by switching network events. *International Journal of Robust and Nonlinear Control*
- [24] **G. Welch ve G. Bishop** (2001). An Introduction to The Kalman Filter. *Course 8, SIGGRAPH 2001*, University of North Carolina at Chapel Hill Department of Computer Science, Chapel Hill, NC, USA
- [25] **M. Lavine** (2007, November). *Introduction to Statistical Thought* (pp. 14-22)
- [26] **S. Gordon** (2006). *The Normal Distribution*, Mathematics Learning Centre, University of Sydney
- [27] **E. M. Hernandez ve D. Bernal** (2008). State Estimation in Nonlinear Structural Systems with Model Uncertainties

- [28] **W. Li, Y. Jia ve J. Du** (2017, September) Distributed extended Kalman filter with nonlinear consensus estimate, *The Journal of the Franklin Institute*. (pp. 1-17)
- [29] **P. S. Maybeck** (1979). Stochastic model, Estimation and Control, Volume 1, *Academic Press*, NY, San Francisco, London
- [30] **İ. Çayıroğlu** (2012). Kalman Filtresi ve Programlama, *Fen ve Teknoloji Bilgi Paylaşımı 2012-1, Karabük Üniversitesi*
- [31] **D. J. Simon** (2006, June). Using nonlinear Kalman filtering to estimate signals, *Embedded Systems Design* (pp. 38-53)
- [32] **R. A. Horn ve C. R. Johnson** (1990). Chapter.5, *Matrix Analysis*, *Cambridge University Press*





ÖZGEÇMİŞ



Ad-Soyad :Orhun Esat ŞAFAK
Doğum Tarihi ve Yeri :19/03/1990 – Fatsa/ORDU
E-posta :orhnest@gmail.com

ÖĞRENİM DURUMU:

- **Lisans:** 2013, Ege Üniversitesi, Mühendislik Fakültesi, Makine Mühendisliği
- **Yükseklisans :** (devam), İstanbul Teknik Üniversitesi, Mekatronik Mühendisliği Anabilim Dalı, Mekatronik Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2014-2015 yılları arasında İstanbul Teknik Üniversitesi Denizcilik Fakültesinde SANTEZ (Gemi Ana Makine Dairesi Simulatörü) projesi kapsamında Tasarım Mühendisi olarak çalıştı.
- 2015-2017 yılları arasında Vestel Elektronik A.Ş.'de (Manisa) Uygulama Uzmanı-Mekanik Ürünleştirme Mühendisi olarak çalıştı.
- 2017-2018 (halen) Türk Traktör ve Ziraat Makineleri A.Ş.'de (Ankara) Araç Entegrasyon Mühendisi olarak çalışmaya devam ediyor.