



**GÖRÜNTÜ İŞLEME DESTEKLİ ENTEGRE
FLOTASYON KONTROL SİSTEMİ
TASARIMI**

(Yüksek Lisans Tezi)

Serdar GÜNDEM

Kütahya - 2024

T.C.
KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
İleri Teknolojiler Anabilim Dalı

Yüksek Lisans Tezi

**GÖRÜNTÜ İŞLEME DESTEKLİ ENTEGRE FLOTASYON
KONTROL SİSTEMİ TASARIMI**

Danışman:
Doç. Dr. Ömer KASIM

Hazırlayan:
Serdar GÜNDEM

Kütahya-2024

Kabul ve Onay

KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

İleri Teknolojiler Ana Bilim Dalında, 202250110101 öğrenci numaralı, Serdar GÜNDEM'in hazırlamış olduğu "GÖRÜNTÜ İŞLEME DESTEKLİ ENTEGRE FLOTASYON KONTROL SİSTEMİ TASARIMI" başlıklı yüksek lisans tez çalışması ile ilgili tez savunma sınavı jüri tarafından yapılmış olup ve adayın tezinin OY BİRLİĞİ ile kabul edilmesine karar verilmiştir.

27/06/2024

Tez Jürisi	İmza	
	Kabul	Ret
Doç. Dr. Ömer KASIM (Danışman)		
Prof. Dr. Rüştü GÜNTÜRKÜN		
Doç. Dr. Mustafa TOSUN		

Onay

Doç. Dr. Eray ACAR

Enstitü Müdürü

Bilimsel Etik Bildirimi

Yüksek Lisans tezi olarak hazırladığım “*Görüntü İşleme Destekli Entegre Flotasyon Kontrol Sistemi Tasarımı*” adlı çalışmanın öneri aşamasından sonuçlandığı aşamaya kadar geçen süreçte bilimsel etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmamda doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

27/06/2024

Serdar GÜNDEM

Özgeçmiş

Eğitim Durumu

- Lisans : Dumlupınar Üniversitesi Teknoloji Fakültesi Elektrik Elektronik Mühendisliği Bölümü
- Lisans : Gazi Üniversitesi Teknik Eğitim Fakültesi Elektrik Öğretmenliği Bölümü
- Lise : STFA Anadolu Teknik Lisesi / Elektrik Bölümü
- Yabancı Dil : İngilizce / Profesyonel Yetkinlik

İş Denevimi

- Oksitem Teknoloji / Teknik Müdür (2011 – Günümüz)
- Endüstriyel Otomasyon sektöründe faaliyet gösteren özellikle savunma, maden ve enerji sektörlerine yönelik kontrol sistemleri tasarımı ve projelendirmesi yapan bir teknoloji firmasıdır.
- Enkosis Otomasyon / Otomasyon Proje Yazılım (2007- 2010)
- Endüstriyel Otomasyon sektöründe faaliyet gösteren, otomasyon malzemeleri satışı ve makine otomasyonu projelendirmesi yapan bir firmadır.
- Teknomar Medikal / Yazılım (2005- 2007)
- Medikal sektöründe faaliyet gösteren, sterilizasyon cihazları imalatı yapan bir firmadır.

ÖZET

GÖRÜNTÜ İŞLEME DESTEKLİ ENTEGRE FLOTASYON KONTROL SİSTEMİ TASARIMI

GÜNDEM, Serdar
Yüksek Lisans Tezi, İleri Teknolojiler Anabilim Dalı
Tez Danışmanı: Doç. Dr. Ömer KASIM
Haziran, 2024, 92 sayfa

Bu çalışmada, özellikle madencilik endüstrisinde yaygın olarak kullanılan flotasyon sistemlerinin otomasyonuna yönelik olarak, görüntü işleme destekli entegre kontrol sisteminin tasarımı yapılmıştır. Flotasyon yüzdürmeye dayalı bir cevher zenginleştirme yöntemidir ve proste oluşun köpüğün formu flotasyon prosesinin verimli olup olmamasında büyük rol oynar. Çalışmada flotasyon prosesi köpük görüntüsü işlenerek köpük hızı tayin edilmiştir. Flotasyon hücresi içerisinde oluşun köpüğün formunu etkileyen, köpük seviyesi ve hava akış debisi için standart olarak kullanılan klasik PID yöntemi manipule edilerek bulanık mantık temelli bir kontrolör tasarımı yapılmıştır. Nihai olarak köpük hızına bağı olarak proses parametrelerini kontrol eden entegre bir yapı tasarlanmıştır. Tasarım esnasında endüstriyel ürünler ve standartlar kullanılmıştır. Yapılan tasarımın kaynak kodları, gerçek zamanlı olarak çalışan endüstriyel PLC (Siemens S7-1513-1) içerisinde yüklenerek simülasyonu ve doğrulaması gerçekleştirilmiştir.

Anahtar Kelimeler: Bulanık Mantık, Endüstriyel Otomasyon, Flotasyon, Görüntü İşleme, Lucas-Kanade

ABSTRACT**INTEGRATED FLOTATION CONTROL SYSTEM DESIGN
USING COMPUTER VISION****GÜNDEM, Serdar****Master Thesis, Department of Advanced Technologies****Supervisor: Assoc. Prof. Doc. Ömer KASIM****June, 2024, 92 pages**

In this study, an integrated control system supported by image processing was designed, especially for the automation of flotation systems widely used in the mining industry. Flotation is an ore enrichment method based on buoyancy, and form of the froth in the process, plays a major role in whether the flotation process is efficient or not. In the study, the froth speed was determined by processing the flotation process froth image. A fuzzy logic-based controller design has been made by manipulating the classical PID method, which is used as a standard for froth level and air flow rate, affecting the form of the froth formed in the flotation cell. Ultimately, an integrated structure was designed that controls the process parameters depending on the froth speed. Industrial products and standards were used during the design. The source codes of the design were loaded into the industrial PLC (Siemens S7-1513-1), which operates in real time, and simulation and verification were carried out.

Keywords: Computer Vision, Flotation, Fuzzy Logic, Industrial Automation, Lucas-Kanade

ÖNSÖZ

Bilgi birikimini paylaşıp beni yönlendiren, en yoğun zamanlarım da kendisini hatırlatarak, sabrını eksik etmeden, çalışmayı bitirmem konusunda desteğini esirgemeyen danışmanım, değerli hocam Doç. Dr. Ömer KASIM'a öncelikli olarak teşekkür ederim.

Tez jürimde bulunan hocalarım Prof. Dr. Rüştü GÜNTÜRKÜN ve Doç. Dr. Mustafa TOSUN hocalarıma akademik tecrübe desteği ve tavsiyelerinden dolayı ayrıca teşekkür ederim.

Bu çalışmada kahvemi eksik etmeyen ve hep yanımda hissettiğim eşim Dilek'e, çalışmalarım esnasında sabır gösteren/göstermeyen oğlum Süleyman Ege'ye, iş yükümü paylaşarak fedakârlık yapan ortağım/kardeşim Kerem ÇETİNKAYA'ya ve bugünlere gelmem de hiçbir fedakarlıktan kaçınmayarak, her zaman yanımda olan, haklarını hiçbir zaman ödeyemeyeceğim annem ve babama sonsuz teşekkür eder, tezimi onlara ithaf ederim.

Bu tezin hazırlanması ve yazım aşamasında destek olan Oksitem Teknoloji bünyesindeki çalışma arkadaşlarıma, maden sektöründeki deneyimlerini aktaran ve maden tesislerindeki gözlem ve çalışmalarımı organize eden Yılmaz Proses Teknoloji firması ekibine ve çalışma süresince bilgisine başvurduğum, emeği geçen herkese bütün içtenliğimle teşekkür eder, saygı ve şükranlarımı sunarım.

İÇİNDEKİLER

Sayfa

ÖZET	v
ABSTRACT	vi
TABLolar LİSTESİ	xi
ŞEKİLLER LİSTESİ	xii
GÖRSELLER LİSTESİ	xiv
KISALTMALAR	xv
GİRİŞ	1

BİRİNCİ BÖLÜM

FLOTASYON SİSTEMİ

1.1. FLOTASYON TANIMI	3
1.2. FLOTASYON TESİS MİMARİSİ	3
1.3. FLOTASYON DİNAMİKLERİ	4
1.4. FLOTASYON OTOMASYONU	7
1.4.1. Flotasyon Hücresi Enstrumantasyonu.....	7
1.4.2. Flotasyon Kontrol Panelleri	9
1.4.3. Flotasyon Kontrol Sistemleri	11
1.5. MODERNİZASYON	12

İKİNCİ BÖLÜM

GÖRÜNTÜ İŞLEME İLE KÖPÜK HIZI TESPİTİ

2.1. KÖPÜK AKIŞI	15
2.2. GÖRME VE GÖRÜNTÜ ANALIZI	16
2.2.1. Hareket Alanı	16
2.2.2. Optik Akış	17
2.2.3. İllüzyonlar	18
2.2.3.1. Işık Etkisi	18
2.2.3.2. Diyafram/Açıklık (Aperture) Problemi.....	18
2.2.3.3. Berber Direği (Barbar Pole) İllüzyonu.....	19
2.2.4. Diğer Etkiler.....	20
2.3. YAKLAŞIM VE METODLAR	20
2.3.1. Lucas Kanade Metodu Açılımı	21

2.3.1.1. Temel Metodoloji.....	21
2.3.1.2. Lucas-Kanade Yaklaşımı	22
2.3.1.3. Matematiksel Gösterim	24
2.4. YÖNTEM VE ÇALIŞMA	25
2.4.1. Materyal ve Metod	26
2.4.2. Köpük Hızının Tespiti.....	28
2.4.3. Verinin İletimi.....	32
2.4.4. Harici Sistemler	33

ÜÇÜNCÜ BÖLÜM

BULANIK MANTIK KONTROLÖRÜ

3.1. HÜCRE KONTROLÜNÜN TEMELİ VE ZORLUKLARI	35
3.2. TASARIM ORTAMI	35
3.3. KONTROLÖR TASARIMI	39
3.3.1. Bulanık Mantık Yaklaşımı	39
3.3.2. Kural Tablosu	40
3.3.3. Algoritma	41
3.3.4. Yazılım.....	43
3.4. SİMÜLASYON.....	46
3.4.1. Factory IO Tasarımı	46
3.4.2. Klasik PID Kontrolü	47
3.4.3. PID Regülatör Katsayıları.....	50
3.4.4. Simülasyon Sonuçları	52

DÖRDÜNCÜ BÖLÜM

ENTEĞRE SİSTEM TASARIMI

4.1. TASARIMIN TEMELİ	56
4.2. KONTROL FİLOZOFİSİ	57
4.3. AKIŞ ŞEMASI.....	59
4.4. KONTROL SENARYOSU.....	61
4.5. PLC YAZILIMI	62
4.6. KONTROL ARAYÜZÜ	65
4.6.1. Çalışma Modları.....	68

SONUÇ VE DEĞERLENDİRME	70
KAYNAKÇA	72
DİZİN	74



TABLolar LİSTESİ**Sayfa**

Tablo 1.1: 10 m ³ Flotasyon Hücresi Test Değerleri.....	5
Tablo 1.2: Kütle Oranı ve Sülfür Kazanım Test Sonuçları.....	5
Tablo 3.1: Bulanıklaştırma Kural Tablosu.....	41



ŞEKİLLER LİSTESİ

Sayfa

Şekil 1.1: Giriş Değerlerine Bağlı Kütle Oranı ve Sülfür Kazanımı	6
Şekil 1.2: Flotasyon Hücresi Enstrumantasyonu	8
Şekil 1.3: Hava Debisi Ölçüm Enstrumanları Tipik Projesi.....	9
Şekil 1.4: Köpük Seviyesi Sensör Bağlantısı Tipik Projesi.....	10
Şekil 1.5: Köpük Seviyesi Kontrol Vanası Tipik Projesi	10
Şekil 1.6: Flotasyon Kontrol Paneli Görünümü	11
Şekil 2.1: Nesne Takibi.....	16
Şekil 2.2: Hareket Vektörleri	17
Şekil 2.3: Optik Akış ve Işık Etkisi	18
Şekil 2.4: Diyafram Açıklık Problemi	19
Şekil 2.5: Berber Direği İllüzyonu.....	19
Şekil 2.6: Yer Değiştirme ve Zaman	22
Şekil 2.7: Flotasyon Hücresi Kamera Konumu	25
Şekil 2.8: Flotasyon Görüntüleri	26
Şekil 2.9: VS Code Geliştirme Ortamı	27
Şekil 2.10: Köpük Hızının Bulunması, Yazılım Aşaması 1	28
Şekil 2.11: Köpük Hızının Bulunması, Yazılım Aşaması 2	29
Şekil 2.12: Materyal 1, Köpük Hızı ve Akışı	30
Şekil 2.13: Materyal 2, Köpük Hızı ve Akışı	30
Şekil 2.14: Materyal 3, Köpük Hızı ve Akışı	31
Şekil 2.15: Materyal 4, Köpük Hızı ve Akışı	31
Şekil 2.16: pyModbusTCP Kütüphanesi	32
Şekil 2.17: ModbusTCP İletişimi Yazılım Aşaması	33
Şekil 3.1: TIA Portal Yazılım Geliştirme Ortamı.....	36
Şekil 3.2: Factory IO OPC Server Konfigürasyonu	38
Şekil 3.3: TIA Portal Simülasyon Konfigürasyonu.....	38
Şekil 3.4: TIA Portal Standart PID bloğu.....	42
Şekil 3.5: Bulanık Mantık Temelli PID_Regulator Bloğu	43
Şekil 3.6: PID_Regulator Değişken Listesi	44
Şekil 3.7: PID_Regulator Yazılım Aşaması 1	45
Şekil 3.8: PID_Regulator Yazılım Aşaması 2	45

Şekil 3.9: PID_Regülator Yazılım Aşaması 3	46
Şekil 3.10: Factory IO Flotasyon Tesis Modeli.....	47
Şekil 3.11: PID Parametreleri (P:1, I:1, D:0) Kontrol Salınımı	49
Şekil 3.12: PID Parametreleri (P:5, I:2, D:0,1) Kontrol Salınımı	49
Şekil 3.13: PID Parametreleri (P:5, I:10, D:0,1) Kontrol Salınımı	50
Şekil 3.14: PID_Regülator Çalışma Parametreleri	52
Şekil 3.15: PID Regulator ile Klasik PID Arasındaki Fark.....	52
Şekil 3.16: Flotasyon Hücreleri Arasındaki Kontrol Salınımı.....	53
Şekil 3.17: Kontrolcülerin Çıkış Enstrümanına Etkisi	54
Şekil 4.1: Kontrol Sistemi Topolojisi	58
Şekil 4.2: Entegre Kontrol Sistemi Akış Şeması.....	60
Şekil 4.3: Köpük Hızı Bloğu Yazılım Değişkenleri	63
Şekil 4.4: Köpük Hızı Bloğu Yazılım Aşaması 1	64
Şekil 4.5: Köpük Hızı Bloğu Yazılım Aşaması 2.....	65
Şekil 4.6: Kullanıcı Arayüzü Ana Sayfa Tasarımı	66
Şekil 4.7: Kullanıcı Arayüzü Ayarlar Sayfası Tasarımı	67
Şekil 4.8: Kullanıcı Arayüzü Alarm Sayfası Tasarımı	68


GÖRSELLER LİSTESİ**Sayfa**

Görsel 1.1: Flotasyon Hücresi.....	3
Görsel 1.2: Flotasyon Bankı.....	4
Görsel 2.1: Kamera ve Sahne Farklılıkları.....	20
Görsel 3.1: S7-1513 PLC Masaüstü Deney Seti	36
Görsel 3.2: Factory IO, Enstrümantasyon Modellemesi	37
Görsel 3.3: Flotasyon Tesisi Simülasyon Arayüzü	39
Görsel 4.1: Operatör ve Kontrol Odası	57



KISALTMALAR

ANSI	American National Standary Institue
CV	Computer Vision
FTE	Fault Tolerant Ethernet
HMI	Human Machine Interface
ISA	Instrument Society of America
PID	Proportional Integrative Derivative
PLC	Programmable Logic Controler
SCADA	Supervisory Control And Data Acquisition





TEZ METNİ

GİRİŞ

Flotasyon yüzdürmeye dayalı bir yöntemdir. Cevher kazanımı sırasıyla, flotasyon hücreleri, bu hücrelerin dizilimiyle flotasyon bankları ve bankların dizilimi ile flotasyon tesisleri sayesinde yapılır.

Flotasyon işleminde köpük yüzeyinin görüntüsü ile performansı arasındaki ilişkilerin proses kontrolünde deneyimli operatörler tarafından kullanıldığı bilinmektedir. Son yıllarda dijital kamera ve yüksek kapasiteli bilgisayarların gelişmesine bağlı olarak, görüntü analiz sistemlerinin flotasyonda hücre bazlı otomatik kontrol sistemi olarak kullanılmalarına yönelik araştırmalar yoğun olarak yapılmaktadır (Zafir ve Ayşe, 2006).

Bu çalışma da flotasyon köpük görüntüsü, görüntü işleme teknikleri kullanılarak değerlendirilmiş, bu işlem sonrası elde edilen veri kullanılarak, köpük hızına bağlı otomatik olarak çalışan entegre bir kontrol sistemi tasarlanmıştır.

Çalışmanın 1. Bölümünde; flotasyon sistemleri, flotasyon dinamikleri ve flotasyon sistemlerinin kontrolüne yönelik otomasyon sistemleri hakkında bilgiler verilmiştir.

Çalışmanın 2. Bölümünde; görüntü işleme konusu üzerine değinilmiş, görüntü işleme konusu ile ilgili bilgiler aktarılmış, bununla beraber köpük görüntüsü üzerinden Lucas-Kanade Metodu kullanılarak köpük hızının ölçümü yapılmıştır. Köpük hızının tespiti yapılırken kullanılan materyal ve metotlara değinilmiş, köpük hızının ölçümüne yönelik çıktılar verilmiştir.

Çalışmanın 3. Bölümünde; günümüz endüstriyel kontrol sistemlerinde son derece yaygın olarak kullanılan PID kontrolcüsüne bağlı ve bu bloğun çıkışına etki ederek kontrol sistemi salınımlarını azaltan, bulanık mantık temelli yeni bir kontrolcü tasarlanmıştır. Tasarımı yapılan kontrolcünün performansı denenmiş ve kıyaslamalı olarak çıktıları verilmiştir.

Çalışmanın 4. Bölümünde; daha önceki bölümlerde, sistemin birer parçası olarak çalışması hedeflenen sistemler birleştirilerek ve nihai kontrol sistemi tasarımına yönelik çalışmalar yapılarak, entegre flotasyon kontrol sistemi tasarımı tamamlanmıştır.



BİRİNCİ BÖLÜM
FLOTASYON SİSTEMİ

1.1. FLOTASYON TANIMI

Flotasyon yüzdürmeye dayalı bir zenginleştirme yöntemidir. Bu yöntemde ayırt edilecek cevherin suya yatkınlığı belirlenir. Bunun yanında cevherin suya karşıtlığı da ölçülür. Özellikle sıvı içerisinde kabarcık oluşumu ile cevherin suda yüzmesi ya da suda batması ile ilgili alanını oluşturan cevherin diğer malzemelerden ayrılması sağlanır. Özellikle endüstriyel üretimde yüzdürme köpüğünün formu oldukça önemlidir. Köpüğün formunun stabil şekilde tutulması için proses kontrol sistemlerinden faydalanılmaktadır. Kullanılan kontrol sistemi, enstrumantasyon ve yazılım ile yüzdürme işlemi optimize edilerek cevher kaynakları daha verimli elde edilmektedir. İyi bir ölçüm ve özellik çıkarımı, verimin daha iyi olması için önemlidir.

1.2. FLOTASYON TESİS MİMARİSİ

Flotasyon tesislerinde kullanılan temel ekipmanlara Flotasyon Hücresi ya da Flotasyon Selülü denilmektedir. Görsel 1.1’de örnek bir flotasyon hücresi gösterilmiştir.

Görsel 1.1: Flotasyon Hücresi



Kaynak: (www.ypt.com.tr, 2024).

Flotasyon hücreleri üzerinde bulunan elektrik motoru ve motora bağlı olan karıştırıcı sayesinde çalışmaktadır. Hücre içerisindeki karışıma mil içerisinden hava gönderilmektedir, böylece köpük oluşumu sağlanır. İçerisinde bulunan karışım köpürdüğü, köpükler üst kısımda birikmekte ve hücreye yeni karışım akışı oldukça seviye kontrolü yapılarak köpüklerin taşması sağlanmaktadır. Karışım akışı diğer bir

hücre ile bağlantılıdır. Böylece karışım sıralı olarak her hücrede prosese tabi tutularak cevher kazanımı elde edilir. Flotasyon hücrelerinin birbirleri ardına yerleştirilerek elde edilen yapıya Flotasyon Bankı adı verilir. Görsel 1.2’de örnek bir flotasyon bankı gösterilmiştir.

Görsel 1.2: Flotasyon Bankı



Kaynak: (ww.ypt.com.tr, 2024).

Flotasyon hücreleri eğimli bir şekilde yana yana dizilirler, hali ile hazırlanan karışım bank boyunca yukarıdan aşağıya doğru bir akış halindedir. İlk hücrenin önünde, flotasyon prosesine tabi tutulacak karışım bulunur, bu karışımın içerisine kazanımı yapılacak cevherin niteliğine göre kimyasal katkıları eklenir ve hazırlanır. Tesisin hacmine ve boyutuna göre hücre ve bank sayısı tasarımı yapılarak, flotasyon tesisleri kurulur.

Flotasyon tesisleri bir çok hücre ve banktan oluşsa da temel olarak kontrol, flotasyon hücresi bazında yapılır, proses esnasında oluşan; köpük miktarı, köpüklerin homojen boylarda olması/olmaması, köpük yönü, köpük hızı, köpük rengi ve köpüklerin kaybolmadan/patlamadan prosese kazandırılması gibi etkenler proses verimliliğini etkiler, proses operatörleri tarafından tecrübesel veriler ile değerlendirilir, ve her hücrenin hava debisi ve köpük seviyesi kontrol edilerek istenilen miktarda köpük kazanımı sağlanır.

1.3. FLOTASYON DİNAMİKLERİ

Flotasyon tesislerinde her hücrenin kendi içerisinde kapalı çevrim olarak kontrolünün sağlandığından daha önce bahsedilmiştir. Bu kontrol temel olarak hava debisi

ve köpük seviyesinin ayarlanması ile yapılır, fakat bu iki önemli kontrol parametresi için, operatörler köpük formlarına bakarlar ve köpüğün formuna göre bu değerlerin kontrol parametrelerini ayarlarlar. Diğer bir bakış açısı ile bu iki parametrenin optimum noktada ayarlanması durumunda da köpük formu düzeltilmektedir ve cevher kazanımı istenilen seviyede olmaktadır. Bu iki önemli değişkenin flotasyon performansına olan etkileri ile ilgili yapılan endüstriyel düzeyde bir çalışma (Ostadrhimi vd., 2022) ve ilgili sonuçları yorumlanacaktır.

Yapılan çalışma da 10 m³ bir flotasyon hücresine aşağıdaki tabloda verilen şekilde hava debisi ve köpük seviyesi kısıtları konularak dokuz aşamalı deneysel bir çalışma gerçekleştirilmiş ve Sülfür cevherinin kazanımı ile ilgili bulgular yorumlanmıştır. Buradaki değerler 10 m³ lük endüstriyel düzeydeki flotasyon hücresinin limitleri dahilinde belirlenmiştir.

Tablo 1.1: 10 m³ Flotasyon Hücresi Test Değerleri

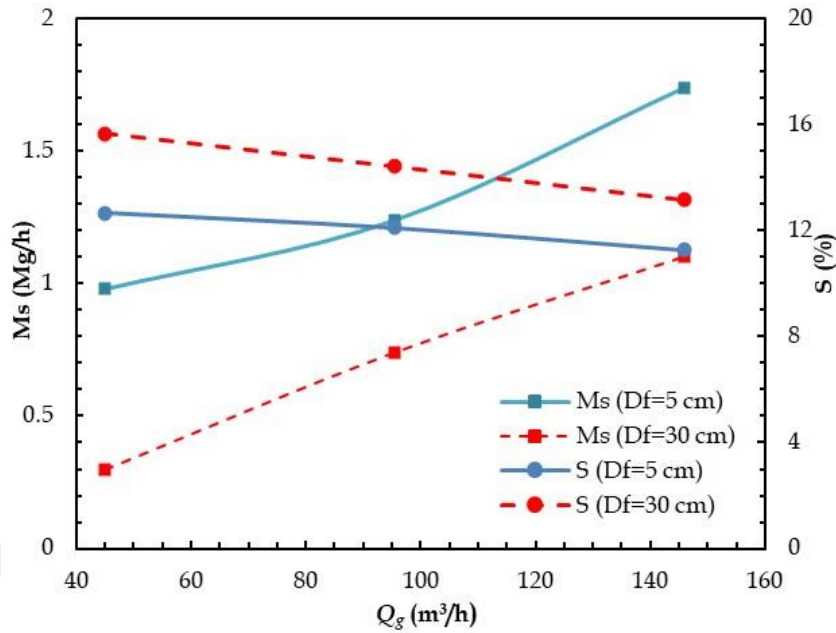
Değerler	Aşama		
	-1	0	+1
Hava Debisi - Q_g (m ³ /h)	45	95.5	146
Köpük Derinliği - D_f (cm)	5	17.5	30

Flotasyon hücresinin en alt ve en üst sınırlarda olan çalıştırma senaryoları sırasıyla uygulanmış ve her aşamadaki cevher kazanımı tablo haline getirilmiştir. Tablo 1.2'de görüldüğü gibi köpük seviyesi arttıkça flotasyon hücresinden çıkan köpük miktarı azalmakta ancak buna karşılık kalitesi artmaktadır. Yani köpük seviyesinin artmasıyla birlikte sülfür minerallerinin konsantreye giren toplam parçacıklara oranının arttığı söylenebilir. Buna karşılık, hava akış hızı arttığında, parçacıkların konsantreye girme hızı da artar, ancak sülfür derecesi daha düşük olur.

Tablo 1.1: Kütle Oranı ve Sülfür Kazanım Test Sonuçları

No.	D_f (cm)	Q_g (m ³ /h)	S (%)	Köpük Miktarı (MS) (Mg/h)
1	5	45	12.67	0.98
2	17.5	45	14.07	0.57
3	30	45	15.67	0.30
4	5	95.5	12.1	1.24
5	17.5	95.5	13.42	0.95
6	30	95.5	14.42	0.74
7	5	146	11.26	1.74
8	17.5	146	12.78	1.48
9	30	146	13.15	1.10

Şekil 1.1: Giriş Değerlerine Bağlı Kütle Oranı ve Sülfür Kazanımı



Veriler yorumlanacak olursa, düşük köpük seviyesinde artan hava debisi flotasyon konsantrisi geri kazanımı üzerindeki etkisinin, yüksek köpük seviyesine göre daha önemli olduğu sonucuna varılabilir.

Bununla birlikte daha yüksek hava debisinin kabarcık yükünün azalmasıyla sonuçlandığı (Ostadrahimi vd., 2019) ve bunun da daha zayıf kabarcıkların oluşumuna yol açtığı bilinmektedir. Bu çalışmada hava debisi miktarı 45 m³/saat'ten 146 m³/saat'e çıktığında kabarcıkların birleşme ve patlama süresi 9 saniyeden 5 saniyeye düşmüştür. Başka bir deyişle, kabarcıkların birleşme oranı yaklaşık %80 oranında artmıştır ve bu da flotasyon konsantrisi geri kazanımı üzerinde olumsuz bir etki göstermiştir.

Flotasyon prosesi performansında köpük bölgesinin büyük önem taşıdığı bilinmektedir. Ancak köpük bölgesinin cevher kazanımı üzerinde hem olumlu hem de olumsuz etkileri olabilir. Köpük bölgesindeki kabarcık hareket kuvveti, bağlı parçacıkları yukarı doğru yüzdürmeye çalışır. Aynı zamanda yerçekimi parçacıkların aşağıya doğru hareket etmesine neden olur. Bu partiküller köpük bölgesinde daha uzun süre kalabilirlerse yer çekiminin etkisi baskın hale gelebilir ve dolayısıyla zayıf bir şekilde tutunan partiküller kabarcıklardan ayrılarak kaybolurlar. Dolayısıyla köpük seviyesinin artırılması ve hava debisinin azaltılmasıyla köpük tutma süresi artar ve buna bağlı olarak geri düşen parçacıkların miktarı da artar. Köpük seviyesinin artması ve hava debisinin azalmasıyla flotasyon konsantrilerinin arttığı görülmüştür. Ancak aynı zamanda konsantrilerin kütleli akış hızı da azalır.

Tablo 1.2'deki verilere bakıldığında, kazanılan köpük miktarı (MS) sütununun aslında flotasyon hızı ile doğru orantılı olduğu söylenebilir. Diğer bir deyişle köpük yüksekliği sabit tutulup hava debisi artırıldığında ya da hava debisi sabit tutulup köpük yüksekliği düşürüldüğünde flotasyon hızı yani köpüklerin savaklara akma hızı artmaktadır. Fakat flotasyon hızının çok artması % S kazanımını düşürmektedir. Nihai olarak cevher kazanımı köpük yüksekliğinin ve hava debisinin optimum noktada yakalanması ile mümkün olabilecektir. Bu sebeple köpük hızına bağlı kademeli bir kontrol sistemi tasarımının yapılması bu çalışmanın ana motivasyonunu oluşturmaktadır.

Mevcut flotasyon hücrelerinde köpük bölgesinden savaklara akan köpüğün hızı ölçülmemektedir. Operatörler ve işletmeciler bu bölgeyi gözlemleyerek, tecrübesel olarak hava debisi ve köpük seviyesini ayarlamaktadırlar.

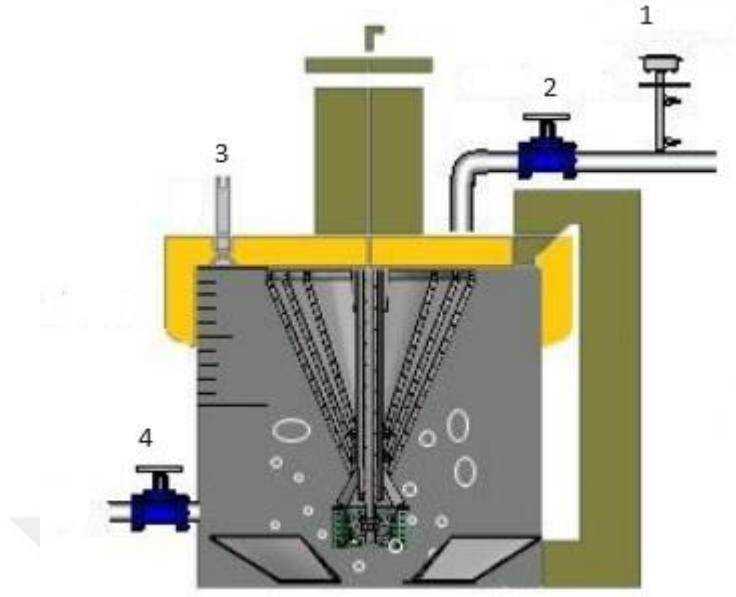
Flotasyon hücrelerinden en üst düzeyde fayda sağlanabilmesi için hava debisinin ve köpük seviyesinin ölçümünün doğru yapılması buna bağlı olarak gelişmiş bir algoritma ile doğru kontrol edilmesi oldukça önemlidir. Bu değişkenlerin doğru ölçülerek doğru kontrol edilebilmeleri için sadece doğru ürün seçimi yeterli olmamak ile birlikte, kullanılan ürünlerin doğru şekilde montaj yapılması ve doğru sistemlere entegre edilmesi gerekmektedir. Bu sebeple devam eden başlıklarda flotasyon sistemlerinin kontrol ve otomasyonu ile ilgili kısa bilgiler verilerek, çevre birimlerin tasarım kriterleri belirlenecektir.

1.4. FLOTASYON OTOMASYONU

1.4.1. Flotasyon Hücresi Enstrumantasyonu

Flotasyon hücrelerinde temel olarak; hücre içerisine gönderilen havanın miktarı ölçülür ve oransal bir vana ile hava miktarı ayarlanır. Aynı zamanda köpük seviyesi ölçülerek hücre çıkışında bulunan oransal bir vana ile seviye kontrolü yapılır. Şekil 1.2'de genel bir enstrumantasyon gösterimi yapılmıştır.

Şekil 1.2: Flotasyon Hücresi Enstrumantasyonu



- Hava debisi ölçümü;

Blower lar tarafından hazırlanan şartlandırılmış kuru hava flotasyon hücrelerine gönderilir, her flotasyon hücresinin girişinde hava hattı bulunmaktadır, hava hatları üzerine daldırma tip termal debimetreler konumlandırılarak flotasyon hücresine giren hava debisi ölçülür.

- Hava debisi kontrolü:

Flotasyon hücresine girecek olan hava miktarı, pnömatik aktüatörlü oransal kelebek vana ile yapılmaktadır.

- Köpük seviyesi ölçümü;

Flotasyon hücresi içerisinde oluşan köpük karışımın üzerine doğru yükselmektedir. Hücre içerisinde bulunan köpük seviyesini karşımdan ayırarak ölçmek için yüzen şamandıralı özel mekanik bir düzenek kullanılmaktadır. Bu mekanik düzeneğin üst kısmında bir plaka bulunur. Flotasyon hücresi üzerine, bu plakayla dik şekilde hizalanmış temassız seviye sensörü takılır. Temassız seviye sensörleri Ultrasonik ya da Radar tip olarak seçilirler.

- Köpük Seviyesi Kontrolü;

Flotasyon hücrelerindeki köpüğün seviyesi köpüğün altında bulunan karışımın miktarı ayarlanarak sağlanır. Hücresinin alt kısmında bulunan Pnömatik aktüatörlü oransal

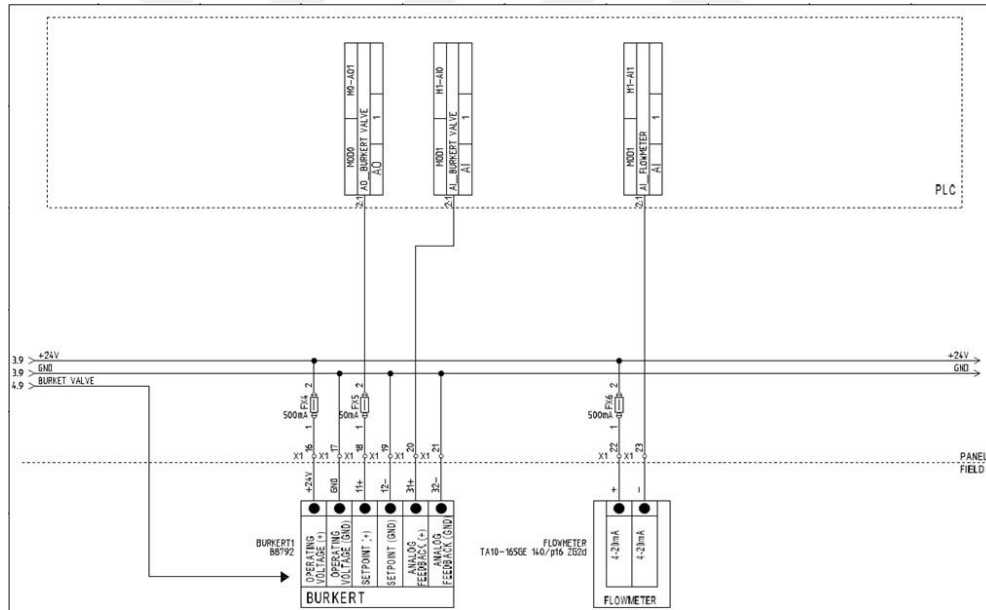
vana ile flotasyon hücresinden kaçırılan karışımın miktarı hali köpüğün seviyesi ayarlanmış olur.

1.4.2. Flotasyon Kontrol Panelleri

Flotasyon hücrelerinde kullanılan ölçüm ve kontrol enstrumanlarının, kablolar vasıtası ile otomasyon panellerine bağlantıları yapılır, otomasyon panelleri tesisin ölçeğine göre tasarlanırlar. Bu doğrultuda kullanılması düşünülen PLC nin giriş çıkış modüllerinin sayısı, panellerin boyutu, kontrolcünün işlem kapasitesi gibi unsurlar değerlendirilir. Tipik devre çizimleri kullanılarak gerekli sayıdaki enstrumanlara göre, panel sayıları belirlenir ve imal edilerek saha montajları yapılır.

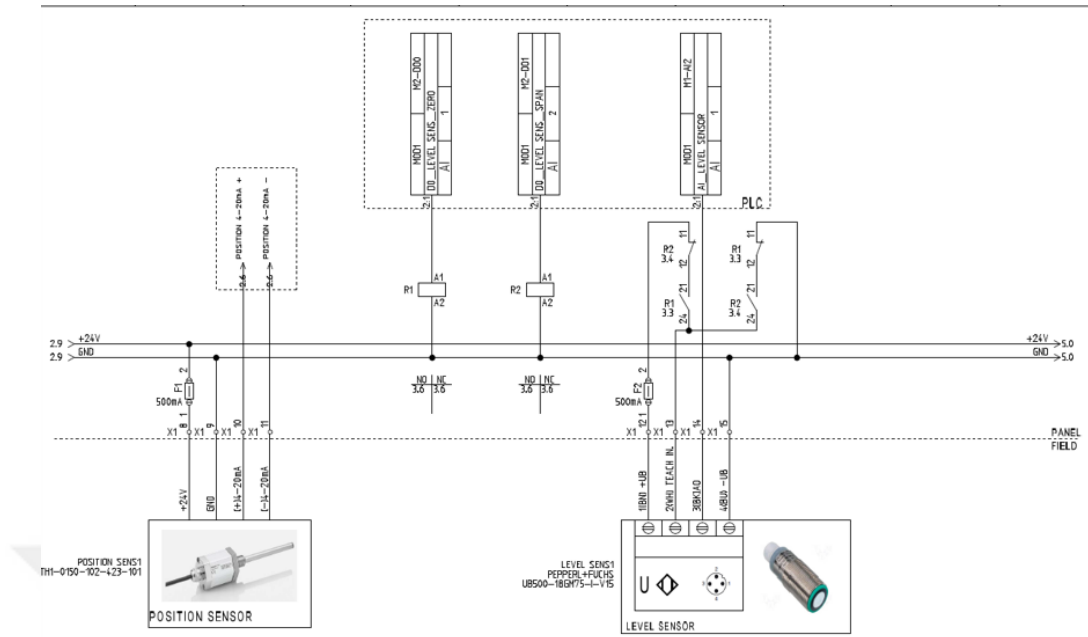
Hava debisi ölçümü ve kontrolü için kullanılacak ekipmanların, kontrol paneli üretimi esnasındaki tipik proje örneği Şekil 1.3’de gösterilen projede verilmiştir.

Şekil 1.3: Hava Debisi Ölçüm Enstrumanları Tipik Projesi

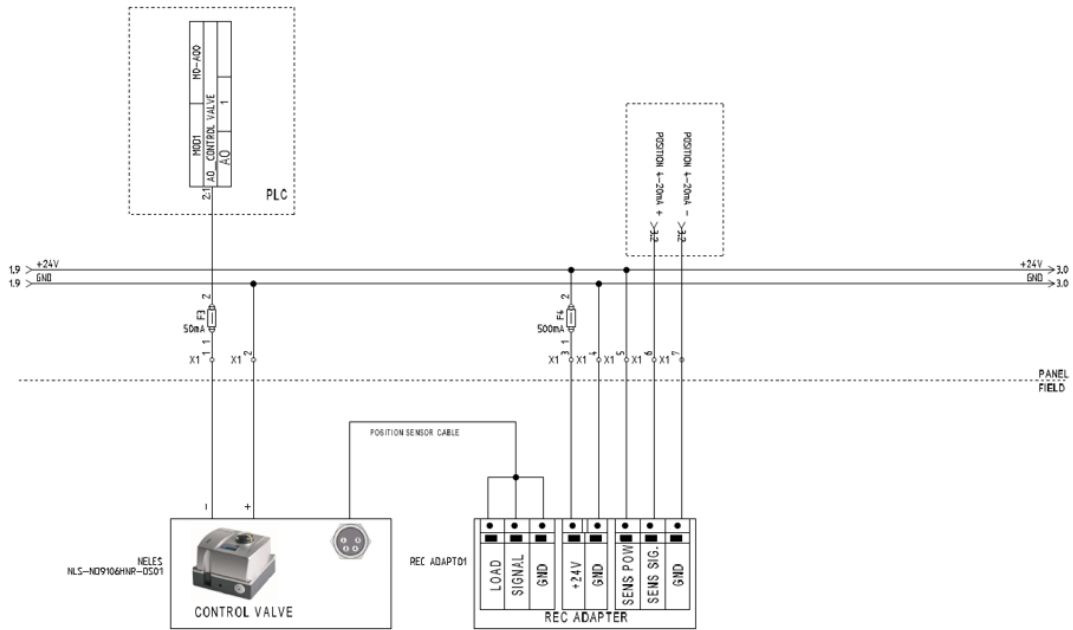


Köpük seviyesi ölçümü ve kontrolü için kullanılacak ekipmanların, kontrol paneli üretimi esnasındaki tipik proje örnekleri sırası ile Şekil 1.4’de ve Şekil 1.5’de gösterilmiştir.

Şekil 1.4: Köpük Seviyesi Sensör Bağlantısı Tipik Projesi

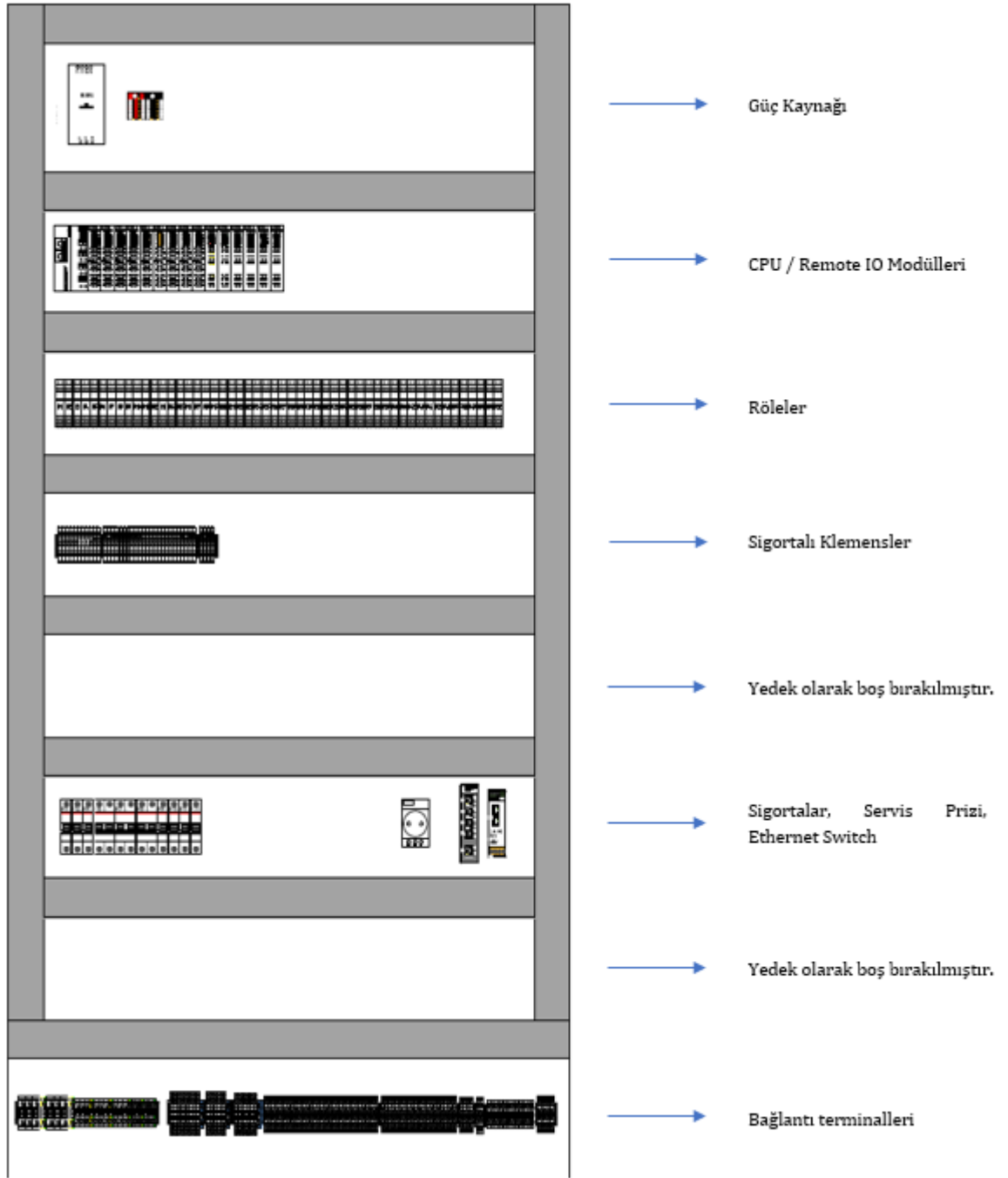


Şekil 1.5: Köpük Seviyesi Kontrol Vanası Tipik Projesi



Tipik devre şemaları kullanılarak imalatı yapılan kontrol panellerinin genel görünümü Şekil 1.6'da gösterilmiştir.

Şekil 1.6: Flotasyon Kontrol Paneli Görünümü



1.4.3. Flotasyon Kontrol Sistemleri

Sistem mimarisi, PLC, uzak IO modülleri, yazılım lisansları, enstrümanlar, vanalar bütününden oluşmaktadır. Paneller, uzak IO Modülleri ile enstrüman ve vanaların bağlantıları, tasarlanan topolojiye uygun saha yerleşimi ve bağlantıları ile yapılır.

Kurulacak sistemin hücre ve bank sayıları, tesis bünyesinde bir üst kontrol sisteminin bulunması/bulunmaması ve son kullanıcı taleplerine göre esnek bir sistem topolojisi tasarımı oluşturulmaktadır.

Sistem topolojisi oluşturulurken, star ya da ring topoloji kullanılabilir. Ring topoloji kullanılarak sistemin iki yönlü haberleşmesi ve yedekleme amaçlanır.

Haberleşme altyapısı olarak ethernet protokolü kullanımı yaygındır. EthernetIP/ ProfiNET/ ModbusTCP haberleşme protokolleri ile sistem tasarlanabilmektedir. Server ile CPU bağlantısı ise FTE (Fault Tolerant Ethernet) ya da yönetilebilir switchler ile yapılabilmektedir.

PLC içerisinde koşturulan yazılım, flotasyon hücrelerinin her birinin kendi içerisinde, hava debisi ve köpük seviyesinin kontrolünü gerçekleştirir. Bununla beraber flotasyon hücrelerinin birbirleri arasındaki iletişimini sağlar, gerektiğinde alarmları ve interlock ları kontrol eder. Bu çalışma için hazırlanan ve PLC içerisinde koşturulan yazılım ve tasarımı hakkındaki detaylar ileri ki bölümler de detaylı olarak aktarılacaktır.

Flotasyon sistemlerinin kontrolü bazı tesislerde hala manuel olarak yapılırsa da, genel olarak PLC tabanlı endüstriyel kontrol sistemleri ile idame ettirilmektedirler. PLC tabanlı kontrol sistemleri, operatörlerin proses parametrelerini değiştirebildikleri HMI (Human Machine Interface) ya da SCADA (Supervisory Control and Data Acquisition) kullanıcı arayüzlerine sahiptirler, operatörler deneyimleri ve bilgileri doğrultusunda proses parametrelerini değiştirerek tesisleri işletirler.

1.5. MODERNİZASYON

Flotasyon dinamikleri bölümünde aktarılan köpük bölgesi hızının ölçümü için, PLC ler ile yapılan mevcut kontrol sistemleri yetersiz kalmaktadır. Bu sebeple mevcut kontrol sistemlerine entegre olabilecek ve köpük hızının ölçülmesini sağlayacak bir ek sistemin kullanılması ve yukarıda belirtilen endüstriyel haberleşme protokollerinden birisi ile, kontrol sistemlerine bağlantısının sağlanması fikri değerlendirilmiştir. Köpük hızına bağlı olarak geliştirilecek bir yazılım algoritması ile flotasyon sistemlerinden optimum noktada verim sağlanabileceği öngörülmüştür.

Endüstriyel sistemler ve dünyadaki gelişmelere bakıldığında flotasyon sistemlerindeki köpük formunun değerlendirilmesi için endüstriyel kameraların ve görüntü işleme algoritmalarının kullanıldığı bilinmektedir.

Bu çalışmada görüntü işleme teknikleri kullanılarak flotasyon hücresindeki köpük akışının hızı tespit edilecektir. Hızın tespit edilmesi sonrası flotasyon hücresinin kontrolünü sağlayan PLC için girdi olarak kullanılacaktır.

İleriki bölümler de sırasıyla; görüntü işleme teknikleri ile hızın tespit edilmesi, bulanık mantık temelli gelişmiş bir kontrol algoritmasının tasarımı ve nihai olarak bu tasarımların birleştirilerek ve kullanılan hız bilgisi ile entegre bir flotasyon kontrol sistemi tasarımı aktarılacaktır.





İKİNCİ BÖLÜM
GÖRÜNTÜ İŞLEME İLE KÖPÜK HIZI TESPİTİ

2.1. KÖPÜK AKIŞI

Flotasyon sistemlerinde proses esnasında oluşan; köpük miktarı, köpüklerin homojen boylarda olması/olmaması, köpük yönü, köpük hızı, köpük rengi ve köpüklerin kaybolmadan/patlamadan proseze kazandırılması gibi etkenler flotasyon prosesindeki verimliliği doğrudan etkilediği bir önceki bölümde aktarılmıştı. Köpük formu ile ilgili birçok parametre değerlendirilebilir olsa da bu bölümde köpük hızının tespitine yönelik bir metod olarak köpük akışının ölçümü konusunda Lucas-Kanade Yönteminin uygulaması gerçekleştirilecek ve köpük hızının tespiti yapılacaktır.

Flotasyon sistemleri sahada aktif olarak çalışan ve geri planda ciddi bir mühendislik birikimi gerektiren özel bir proses olarak karşımıza çıkmaktadır. Her cevherin kazanımı farklı şekillerde ve farklı uzmanlıklar ile gerçekleşmektedir. Bu sebeple problemin tanımını yaparken pratik ve teorik olarak ayırmak uygun olacaktır.

Flotasyon prosesinde köpük akışının doğru analiz edilememesi prosesin verimliliğini ve cevher kazanımını doğrudan etkilemektedir. Köpük akışının analizi ve yorumu bilgisayar destekli görüntü işleme algoritmaları kullanılarak yapılabilmektedir. Bununla ilgili olarak maden sektöründe faaliyet gösteren büyük olarak niteleyebileceğimiz birçok firmanın kendine ait donanım ve yazılım paketleri bulunmaktadır. Ülkemizde ise bu konu ile ilgili somut bir faaliyet ya da çalışma bulunmaması ile birlikte flotasyon sistemlerindeki köpük akışının tespiti büyük çoğunlukla insan odaklı olarak operatörler tarafından yapılmaktadır. Yapılan gözlemler kalitatif nitelikte olup kantitatif bir yaklaşım sergilenmemektedir. Operatörün tecrübesel bilgilerine dayalı olarak proses verileri değerlendirilmekte ve bu tecrübeye istinaden proses parametreleri manuel ya da otomasyon destekli olarak değiştirilmektedir.

Köpük akışının ölçümü noktasında temaslı ve temassız olarak iki yaklaşım sergilenebilir, fakat akışkan ile doğrudan temas eden- yani bir sensör ya da algılayıcı bir sistem ile köpük akışının ölçümü pratikte uygulaması zor olup, bu sebeple tercih edilmemiştir. Bunun yerine flotasyon prosesine hâkim, deneyimli ve köpük analizi konusunda fikir sahibi kişilerin tecrübeleri ve bilimsel veriler doğrultusunda, bilgisayar destekli görme sistemi ve görüntü işleme yazılımları ile bu konunun çözümü gerçekleştirilmiştir. Bu aşamada görüntü işleme algoritmaları üzerinde durulması gerekmektedir. Optik akışın saptanabilmesi için birçok yöntem uygulanabilmektedir fakat optik akışın tespiti her zaman doğru olmayabilir. Sabit olan ve referans noktası

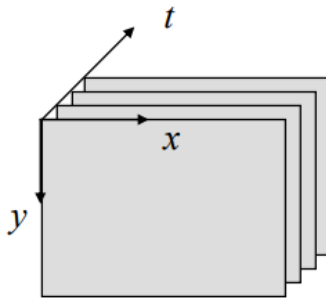
alınabilen tiyatral bir görüntü yerine, sürekli akışın olduğu ve sürekli değişen bir görüntü dizisi üzerinde çalışılması ve bunun için en uygun yöntem ile ilerlenilmesi gerekmektedir.

2.2. GÖRME VE GÖRÜNTÜ ANALIZI

İnsan beyni, gözü kullanarak nesnelere algılamaktadır ve bazen problemler ile karşılaşır. Bu problemlerden biri, üç boyutlu olan verinin iki boyutlu retina üzerindeki izdüşümüdür. Bu problem sebebiyle, iki boyutlu bilgiye bağlı olan doğru görsel izlenimin elde edilmesi için, beyin daha sonra bu işlemi tersine çevirmektedir. Nihai olarak görüntünün elde edilmesi bu şekilde gerçekleşir (Özöntürk, 2007).

Endüstride ise görüntü işleme sistemleri kameralar kullanılarak yapılmaktadır. Kamera göz vazifesi, bilgisayar destekli algoritma ve yazılımlar da beyin vazifesi görürler. Kamera kullanılarak elde edilen ardışık video çerçevelerindeki hareketli bir nesnenin zaman boyunca, çerçeve içerisindeki konum bilgisinin belirlenmesine nesne takip işlemi denir. Şekil 2.1’de nesne takibine ilişkin görsel verilmiştir.

Şekil 2.1: Nesne Takibi



Video kaydı üzerinde gerçekleştirilen bu basit takip işlemi zamanla çok geniş bir kullanım alanına sahip olmuştur (Dilek, 2012). Flotasyon prosesi içerisinde köpük akışının ölçümü de aslında bir nevi takip işlemi olup, takip neticesinde zamana bağlı bir denklem ile köpük akışı ölçümü gerçekleştirilir.

2.2.1. Hareket Alanı

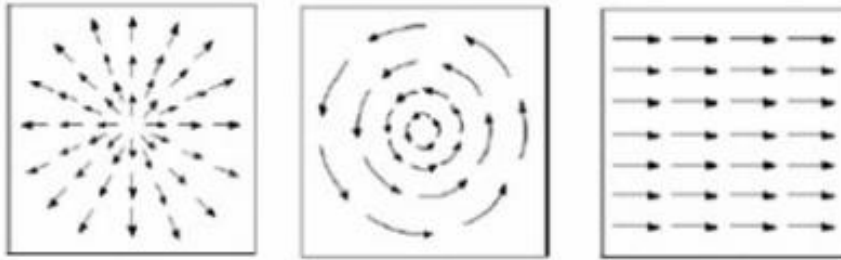
Bilgisayar destekli görmede hareket alanı, üç boyutlu hareketin bir kamera görüntüsüne iz düşürülmesi olarak tanımlanır. Basit bir kamera modelinde, görüntü üzerindeki her nokta $(y1, y2)$, üç boyutlu sahnedeki bazı noktaların izdüşümüdür, fakat uzaydaki sabit bir noktanın iz düşümünün yeri zamanla değişebilir. Bunun neticesinde hareket alanı, görüntü koordinatlarını iki boyutlu bir vektöre eslestiren bir fonksiyon

olarak gösterilebilir. Hareket alanı, iz düşümü alınan üç boyutlu hareketin ideal tanımıdır; bu alan teorik olarak tanımlanabilir fakat pratikte görüntü verisinden sadece bir hareket alanı yaklaşımı elde edilebilir. Basit bir kamera modeli tanımı ile hareket alanını açıklayabiliriz; adresleme fonksiyonları (m_1, m_2) olarak, üç boyutlu uzayda yer alan her nokta da (x_1, x_2, x_3) noktaları ile tanımlanırsa, iki boyutlu bir görüntü noktasına (y_1, y_2) olarak iz düşümü alınmaktadır. Aşağıda verilen fonksiyon hareket alanını tanımlamakta yardımcı olacaktır.

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} m_1(x_1, x_2, x_3) \\ m_2(x_1, x_2, x_3) \end{pmatrix} \quad (2.1)$$

Kamera tarafından resmedilen sahnenin zamana göre değiştiği varsayımı yapılmaktadır; sahne, birbirine göre göreceli olarak hareket eden nesnelere oluşmaktadır (Dilek, 2012). Üç boyutlu noktalar uzayda hareket ederlerken, bunlara karşılık gelen noktalar da hareket eder. Hareket alanı, sahnedeki tüm noktalar için görüntüde gerçekleşen hareket vektörlerinden oluşmaktadır. Bu vektörler kullanılarak hareket hakkında fikir sahibi olunabilir. Örneğin aşağıda verilen Şekil 2.2’de sırasıyla; İleri yönlü hareket, dönme hareketi ve yatay yönde hareket ile ilgili vektörler gösterilmiştir.

Şekil 2.2: Hareket Vektörleri



2.2.2. Optik Akış

Optik akış, bir görüntüdeki parlaklık desenlerinin hareketinin görünür hızlarının dağılımıdır. Optik akış, nesnelere ve izleyicinin göreceli hareketinden kaynaklanabilir (Berthold, Horn, Brian ve Schunck, 1980). Kameralar ile yapılan görüntü işleme prosesinde, hareketin algılanması için sensörler kullanılır ve görüntünün her pikseline karşılık gelen şiddetin genliği belirlenebilir. Bu bilgi sayesinde çerçeve üzerindeki her noktaya bir kantitatif değer atanabilir ve bir sonraki çerçeveler için de aynı yöntem kullanılarak akış tespiti yapılabilir. Fakat her noktanın şiddeti farklı olmayabilir ve aynı

şiddete sahip komşu piksel noktaları da olabilir. Bu durumda optik akışın kusursuz bir şekilde doğrudan ölçülemediği ortaya çıkıyor, bu da bizi bir dizi yöntem ve varsayım yaparak ilerlememiz gerektiği noktasına götürür. Optik akış, görüntüdeki hareketin yön ve hızını tanımlar. Bir genelleme yapılacak olursa, optik akış hareket alanına karşılık gelir fakat bu denklik her durum için geçerli değildir. Hareket alanının farklı, optik akışın farklı olduğu durumlar da gerçekleşebilmektedir bu durumlar görüntü illüzyonu olarak tanımlanabilir.

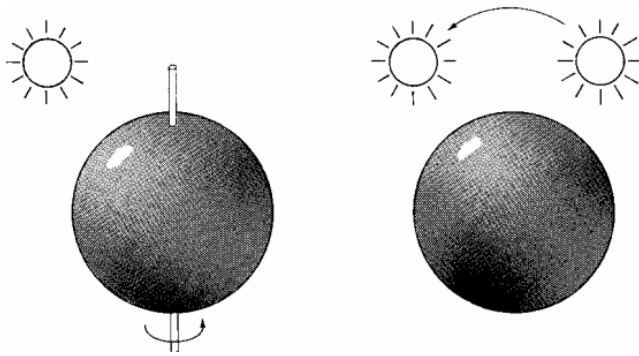
2.2.3. İllüzyonlar

Nadirde olsa görüntülerde oluşan ilüzyon etkisi her nasıl gözümüzü yanıltıyorsa, görüntü işleme esnasında, kamera dan alınan veriyi de yanıltacaktır. Bu durumların gösterilmesi ile hareket alanı ve optik akış kavramları daha anlaşılır olacaktır.

2.2.3.1. Işık Etkisi

Bazı durumlarda ışığın etkisi ve açısı optik akışın tespitini oldukça zorlaştırabilmektedir. Bu duruma bir örnek olarak Şekil 2.3’de gösterilebilir.

Şekil 2.3: Optik Akış ve Işık Etkisi

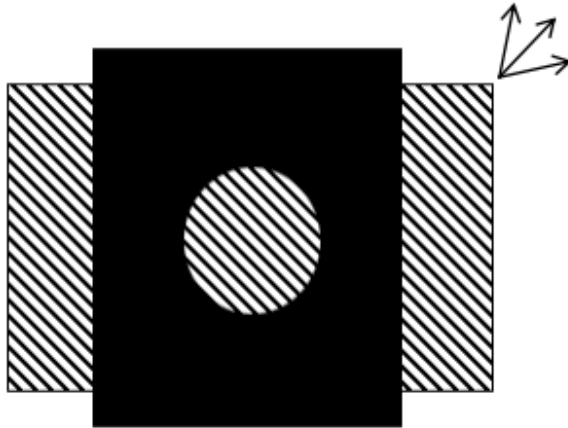


Kendi eksenini etrafında dönen bir küre, sabit bir ışık açısı ve şiddeti ile sanki dönmüyormuş gibi gözükse de aslında hareket halindedir. Bunun tam tersi olarak sabit duran bir kürenin üzerindeki ışık şiddeti ve açısını değiştirdiğimizde hareket olmadığı halde optik bir akış gözlemlenecektir.

2.2.3.2. Diyafram/Açıklık (Aperture) Problemi

Diyafram, açıklık yada aralık problemi olarak nitelendirilen durumdur. Şekil 2.4’te görüldüğü üzere belirli bir aralıktan ve genelde simetrik bir sisteme sahip görüntülere baktığımızda yada bu görüntüleri işlediğimizde karşılaşılan problemdir.

Şekil 2.4: Diyafram Açıklık Problemi

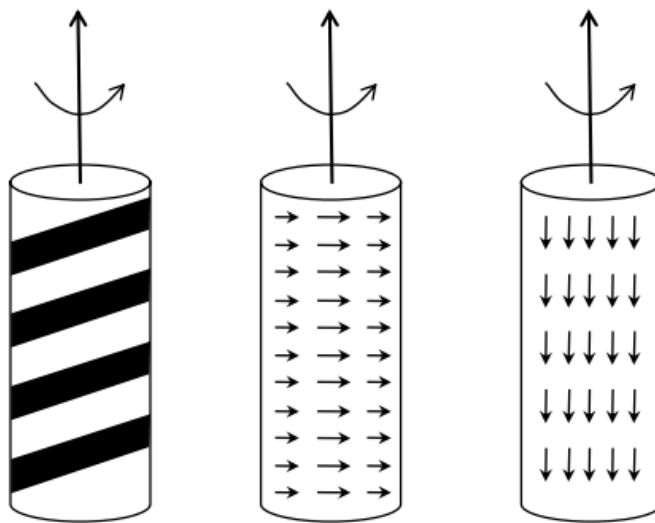


Arka tarafta bulunan levhanın hareket ettirilmesi durumunda optik bir akış ve hareket gerçekleşecektir, fakat bu levhanın hangi yöne doğru hareket ettiği belirlenemeyecektir.

2.2.3.3. Berber Direği (Barbar Pole) İllüzyonu

Belki de hareket alanı ve optik akış kavramlarını anlamak için en uygun örnek olarak nitelendirebileceğimiz ve “Barbar Pole Ilusion” olarak geçen meşhur göz yanılgısı durumudur. Kendi eksenini etrafında dönen, genelde mavi ve kırmızı şeritlerden oluşan bir silindirik yapıdır. Şekil 2.5’te Berber Direği İllüzyonu gösterilmiştir.

Şekil 2.5: Berber Direği İllüzyonu



Silindirik yapı kendi eksenini etrafında dönmektedir ve buradaki hareket sağa doğrudur, ortadaki şekil hareketin yönünü göstermektedir. Fakat gözümüzle ya da bir

kamera vasıtası ile görüntüyü değerlendirdiğimizde hareketin aşağıya doğruymuş gibi olduğunu gözlemleriz, bu ise optik akışın aşağı doğru oluşundan kaynaklanmaktadır. Yani hareket ile optik akış arasında oldukça büyük bir yanlısama bulunmaktadır.

2.2.4. Diğer Etkiler

Görüntünün doğru işlenebilmesi ve tanılanabilmesi için, harici etkenlerin ve etkilerin de minimuma indirgenmesi gerekmektedir. Grösel 2.1’de bu konuya açıklık getirmek için bazı resim örnekleri verilmiştir.

Görsel 2.1: Kamera ve Sahne Farklılıkları



İlk resimde bir manav dükkânı görülmektedir, bu resimde sahne sabit fakat gözlemci/ kamera hareketli durumdadır, ortada bulunan resim de gözlemci/kamera sabit iken sahne hareketli durumdadır, son resimde ise hem sahne hareketli hem de gözlemci/kamera hareketlidir. Tabii olarak tüm bu resim ve örnekler için ışık kaynağının değişken oluşu da görüntü kalitesini ve dolaylı olarak görüntü işleme kullanarak varmaya çalıştığımız çıktıyı etkilemektedir.

2.3. YAKLAŞIM VE METODLAR

Nesne takibi, hız tespiti, görüntü analizi/yorumu gibi farklı alanlardaki, farklı çalışmaların tamamının çözümüne yönelik yaklaşım ve metodların temelinde, optik akışın tespiti bulunmaktadır. Optik akış teknikleri; Fark Teknikleri, Frekans_Temelli Teknikler ve Eşleştirme Teknikleri olarak üç grupta sınıflandırılmışlardır. Bunun dışında Ortalama Değer Kayması Metodu da yaygın olarak kullanılan yaklaşımlardan birisidir.

Ortalama Değer Kayması Metodu istatistiksel bir yöntemdir ve nokta dağılımının en yoğun olduğu yeri bulmayı sağlar. Ortalama değer kayma yöntemi, Fukunaga ve Hostetler tarafından ilk kez ortaya çıkarılmış parametrik olmayan bir öbekleme yöntemidir (Fukunaga ve Hostetler, 1975). Yöntem, Cheng’in makalesine (Cheng, 1995) kadar ilgi görmemiştir ve üstün özelliklerine rağmen istatistiksel literatürde de görülememiştir. Ancak yöntem, daha sonraları kesimleme ve renkli nesne

takibi uygulamalarında kullanılarak tekrar gündeme gelmiştir (Cheng, 1995) (Comaniciu, ve Meer, 1997). Ortalama değer kayması algoritması, hedefin istatistiksel bilgilerine dayanarak renk, doku gibi görsel nitelikleri verimli bir şekilde takip eder. Bu yöntem birden çok (farklı renkli bölgelere sahip) hedefin, örneğin manzarayı, yüzü, insanları, gerçek zamanlı olarak ve hedeflerin kısmen kaybolması durumunda bile sağlam bir takip sağlamaktadır. Aktif görüntüdeki hedefin koordinatı ile sonraki görüntü çerçevesindeki hedefin özelliklerine benzer en yakın bölgenin koordinatının belirlenmesine dayanır.

Fark Teknikleri: Uzay-zamansal şiddet türevlerinden görüntü hızını hesaplar. Optik akış, görüntü dizilerindeki yerel türevler üzerine kurulu yerel görüntü hareketi yaklaşımıdır. İki boyutlu görüntüde yerel türevler hesaplanarak her görüntü pikselinin ardışık görüntüler arasında ne kadar ilerlediği belirlenir. Optik akış, gözlenen görüntülerin uzaysal düzenlemesi ve bu düzenlemenin değişim oranı hakkında yararlı bilgiler vermektedir. Hareket eden görüntüler görüntü parlaklığında zamansal değişimlere neden olmaktadır ve tüm bu zamansal şiddet değişimlerinin sadece harekete bağlı olduğu varsayımı yapılmaktadır (Barron, Fleet ve Beauchemin, 1994). Hangi optik akış metodu kullanılırsa kullanılsın, her zaman görüntü şiddet türevleri hesaplanmak zorundadır. Yaygın olarak Horn-Schunck Optik Akış Modeli ile Lucas-Kanade Optik Akış Modeli kullanılmaktadır.

Frekans-temelli Teknikler: Hız ayarlı filtre çıkışındaki enerji/faz bilgisini kullanırlar. Eşleştirme Teknikleri ise kısıtlı sayıdaki görüntüden değişik görüntü özelliklerini eşleştirerek görüntü yer değişimlerini hesaplar.

Bu çalışmanın da ana konularından bir tanesi olan Lucas-Kanade Metodu ayrıca detaylı olarak aktarılacaktır. Böylece optik akış fark teknikleri konusunda da detaylı bilgi verilmiş olacaktır.

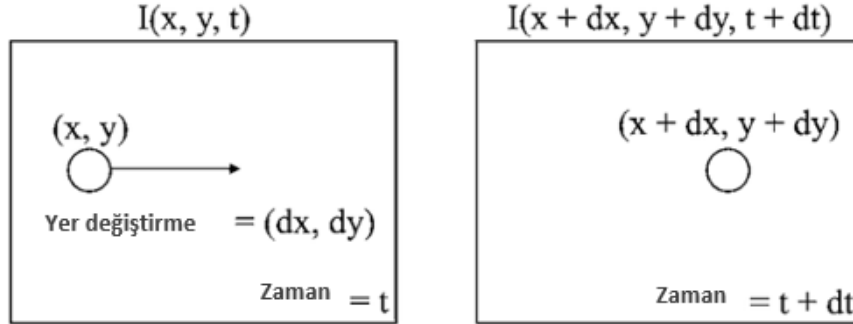
2.3.1. Lucas Kanade Metodu Açılımı

2.3.1.1. Temel Metodoloji

Optik Akış, parlaklık veya yoğunluk sabitliği prensibine bağlı bir tekniktir. Bu prensibe göre bir görüntüdeki pikselin parlaklığı zamanla aynı kalır. Bu piksele anlık olarak bakıldığında (" t " zamanı), diğer piksel aynı hızda olsa bile parlaklığının başka bir pikselin daha sonraki bir zamandaki parlaklığıyla ($t + dt$) aynı olacağı anlamına gelir. Sağ tarafta görülen kare (dx ve dy) ile kaydırılmış noktadır. Optik akıştaki akış alanı, piksel

konumlarındaki bu kaymaları tahmin eder ve bu çıkarım, optik akışın arkasındaki temel varsayımdır.

Şekil 2.6: Yer Değiştirme ve Zaman



Videonun ilk karesindeki bir piksele “t” zamanında bakılır ve bu kare $I(x, y, t)$ olarak tanımlanabilir. Kısa bir süre (dt) sonra çekilen bir sonraki karede piksel belirli bir mesafe (dx, dy) kadar hareket eder. Pikseller tamamen aynı olduğundan ve parlaklıkları değişmediğinden piksel hareket etse bile görünümünün ve parlaklığının aynı kaldığı söylenebilir ve bu aşağıdaki denklem ile ifade edilebilir.

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad (2.2)$$

Sağ taraftaki Taylor serisi yaklaşımını kullanarak ve ortak terimleri çıkarıp basitleştirerek dt 'ye bölünürse aşağıdaki ifadeler elde edilir:

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y} \quad u = \frac{dx}{dt} \quad v = \frac{dy}{dt} \quad I_t = \frac{\partial I}{\partial t} \quad (2.3)$$

Uzamsal türev

Optik akış

Zamansal türev

$$I_x u + I_y v + I_t = 0 \quad (2.4)$$

Az önce elde edilen denklem genel olarak Optik Akış denklemi olarak adlandırılır. Bununla birlikte (u, v) değerleri bilinmiyor. Burada iki bilinmeyenli bir denklem ile karşılaşılır, bu problemin üstesinden gelmek için farklı yöntemler önerilmiştir ve bunlardan biri de Lucas-Kanade metodudur.

2.3.1.2. Lucas-Kanade Yaklaşımı

Lucas-Kanade nin önerdiği metod, bir dizi görüntüdeki nesnelerin, görünen hareketini tahmin etmek için yaygın olarak kullanılan bir yöntemdir. Bruce D. Lucas ve

Takeo Kanade tarafından 1981 tarihli ufuk açıcı makaleleri olan "Stereo Görüş Uygulamasıyla Yinelemeli Görüntü Kayıt Tekniği"nde tanıtıldı (Barron ve Thacker, 2004).

Lucas-Kanade Yöntemi, Optik Akışın hesaplanmasında yaygın olarak kullanılan yaklaşımlardan biri olarak öne çıkar. Optik Akış denklemindeki u ve v değerlerini hesaplamak için iki temel varsayım üzerinde durur;

Akış yerel olarak düzgündür. Bu, görüntünün küçük, lokalize bir bölgesindeki nesnelerin veya noktaların hareketinin nispeten tutarlı olduğu ve anlık değişiklikler sergilemediği anlamına gelir.

Komşu pikseller aynı yer değiştirmeye sahiptir. Başka bir deyişle, seçilen bölgedeki komşu piksellerin benzer harekete sahip olduğu varsayılarak, algoritmanın genel hareketi tek bir vektörle temsil etmesine olanak sağlanır.

Lucas-Kanade Metodu bu varsayımlar kapsamında; yöntem, referans ve hedef görüntüdeki piksellerin yoğunlukları arasındaki hatayı en aza indiren ve en uygun yer değiştirme vektörünü bulmayı amaçlamaktadır.

Lucas-Kanade yöntemi yinelemelidir ve aşağıdaki adımlarla özetlenebilir:

- Ön işleme: Gürültüyü azaltmak ve yöntemin sağlamlığını artırmak için görüntüler genellikle bir Gauss filtresi kullanılarak yumuşatılır. Bu yöntem dışında farklı şekilde filtreler de kullanılabilir.
- Uzamsal hesaplama: Görüntü yoğunluğunun uzamsal gradyanları (I_x , I_y) hesaplanır. Bunlar sırasıyla x ve y yönlerinde görüntü yoğunluğunun değişim hızını temsil eder.
- Zamansal hesaplama: Zamansal gradyan (I_t), ardışık görüntülerdeki karşılık gelen piksellerin yoğunlukları arasındaki fark olarak hesaplanır. Bu gradyan, görüntü yoğunluğunun zaman içindeki değişim oranını temsil eder.
- En küçük kareler tahmini: Yer değiştirme vektörü (u , v), uzaysal ve zamansal gradyanlardan elde edilen doğrusal bir sistemin çözülmesiyle tahmin edilir. Bu, referans ve hedef görüntüler arasındaki karesel farkların toplamını en aza indiren en küçük kareler yöntemi kullanılarak yapılır.
- Yineleme ve iyileştirme: Tahmini yer değiştirme, hedef görüntüyü referans görüntüye doğru çarpıtmak için kullanılır. 2-4 arasındaki adımlar, tahmin edilen

yer deđiřtirmenin yinelemeler arasında önemli ölçüde deđiřmediđi durumlarda meydana gelen yakınsamaya kadar tekrarlanır.

Lucas-Kanade yönteminin büyük yer deđiřtirmelere, dönmeye ve ölçek deđiřikliklerine duyarlılıđı gibi bazı sınırlamaları vardır. Bu sorunları çözmek için yıllar içinde Lucas-Kanade yönteminin çeřitli uzantıları ve varyasyonları önerilmiřtir.

2.3.1.3. Matematiksel Gösterim

Temel denklem;

$$I_x u + I_y v + I_t = 0 \quad (2.5)$$

Olarak tanımlanmıřtı, ve bu denklem ařađıdaki řekilde yazılabilir.

$$I_x u + I_y v = -I_t \quad (2.6)$$

Lucas-Kanade yöntemi, bir videoda nesnelerin nasıl hareket ettiđini anlamak için özel noktalara ihtiyaç duyar. Bu noktaları elde etmenin popüler bir yolu Shi-Tomasi köře tespit tekniđini kullanmaktır. Bu teknik videodaki nesnelerin köřelerini bulur ve daha sonra bu yamalar Lucas-Kanade yöntemiyle hareketi anlamak ve takip etmek için kullanılır. Herhangi bir görüntüde ($n \times n$) piksel ölçülerine sahip bir görüntü yaması (parçası/örneđi) řeklinde ilerlenir ve bunlar ařađıdaki řekilde ifade edilmektedirler,

$$I_x(p1)u + I_y(p1)v = -I_t(p1) \quad (2.7)$$

$$I_x(p2)u + I_y(p2)v = -I_t(p2) \quad (2.8)$$

$$\vdots \quad \quad \quad \vdots$$

$$I_x(pn)u + I_y(pn)v = -I_t(pn) \quad (2.9)$$

Bu denklemler matris formunda temsil edilebilir.

$$\begin{bmatrix} I_x(p1) & I_y(p1) \\ I_x(p2) & I_y(p2) \\ \vdots & \vdots \\ I_x(pn) & I_y(pn) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p1) \\ I_t(p2) \\ \vdots \\ I_t(pn) \end{bmatrix} \quad (2.10)$$

$$\begin{matrix} A & x & b \\ n^2 \times 2 & 2 \times 1 & n^2 \times 1 \end{matrix}$$

Bu gösterim sonrası bilinmeyenlerin hesaplanması için “En küçük kareler yöntemi” kullanılır. Görüntü yaması (P: Patch) içerisindeki, her nokta (p: piksel) aşağıdaki şekilde tanımlanır.

$$\begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum_{p \in P} I_x I_t \\ \sum_{p \in P} I_y I_t \end{bmatrix} \quad (2.11)$$

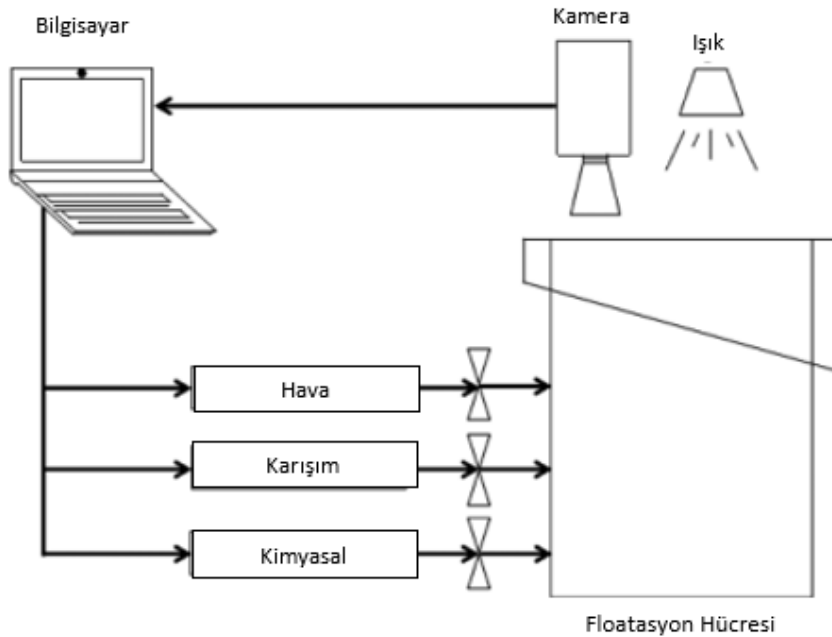
Son olarak, her iki tarafın tersini alarak, u ve v bilinmeyenlerini içeren x vektörü hesaplanabilir ki bu da optik akış vektörüdür (Lucas ve Kanade, 1981).

$$x = (A^T A)^{-1} A^T b \quad (2.12)$$

2.4. YÖNTEM VE ÇALIŞMA

Flotasyon prosesindeki köpük akışı ölçümü için alınan video görüntülerinde gözlemci/ kamera sabit ve sahne hareketlidir. Bununla birlikte sabit bir ışık şiddeti ile görüntü desteklenmiştir. Ayrıca flotasyon prosesinin doğası gereği önceki konularda bahsedilen illüzyon ya da yanılsamalar bulunmamaktadır. Aşağıda verilen Şekil 2.7’de görüntünün alınması için kullanılan yöntem görsel olarak verilmiştir (Bhondayi, 2020). Bununla birlikte farklı materyaller üzerinde de işlem yapılarak bir değerlendirme yapılacaktır.

Şekil 2.7: Flotasyon Hücresi Kamera Konumu



2.4.1. Materyal ve Metod

Verilen bilgiler neticesinde flotasyon prosesinden alınan görüntülerde hareket alanının optik akışa eşit olduğu, en doğru ölçüm koşullarının varsayımı yapılabilir. Bu varsayım sonucunda Lukas Kanade metodunun, köpük akışının ölçümünde uygun bir metod olacağı öngörülmektedir.

Materyal olarak dört adet farklı Şekil 2.8’de verilen video görüntüsü kullanılmıştır. Işık, gölge, açı, renk, çözünürlük gibi hususların farklı olduğu materyaller seçilmiştir. Bahsedilen hususların farklılığı aslında cevherin (madenin) ve proses sahalılarının farklılığını da simüle etmiş olacaktır. Böylece bu etkilerin değerlendirilmesi yapılabilecektir.

Şekil 2.8: Flotasyon Görüntüleri



(1)



(2)



(3)



(4)

İlk video görüntüsü; Feldspat üretim tesisinden alınan bir video görüntüsüdür. Kamera dik olarak 90 derecelik bir açıda ve sabit ışık kaynağı ile desteklenerek çekim yapılmıştır.

İkinci video görüntüsü; Aynı feldspat üretim tesisinden alınan bir video görüntüsüdür. Kamera dik olarak 90 derecelik bir açıda ve ışık kullanılmadan doğal olarak gölgelerin bulunduğu farklı bir flotasyon hücresinde çekim yapılmıştır.

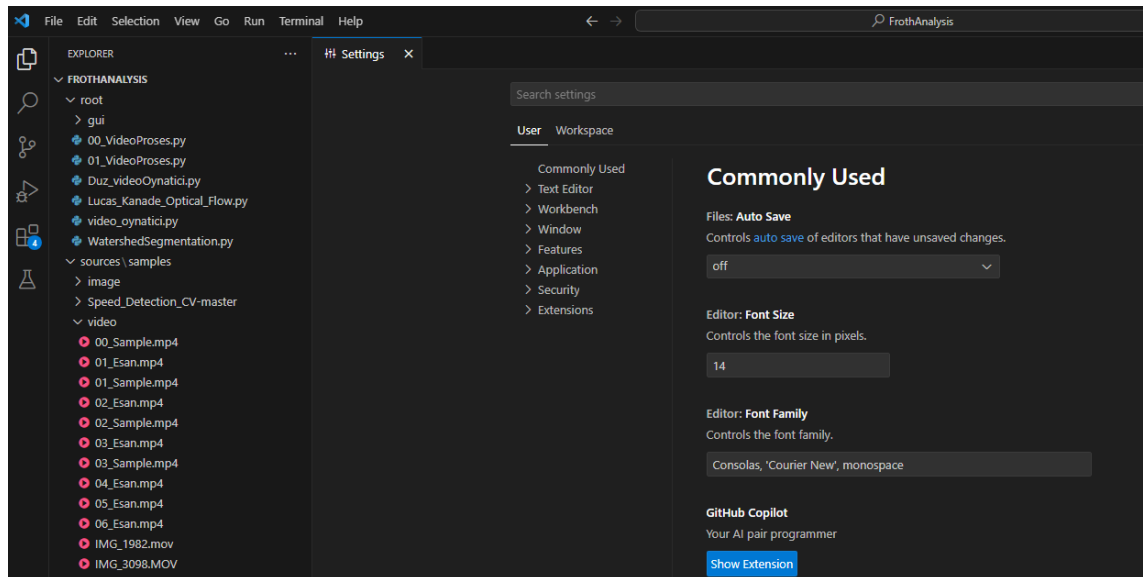
Üçüncü video görüntüsü; internet ortamından temin edilmiştir, kurşun-çinko üretimi yapılan bir tesiste çekilen video görüntü dosyası olup sabit ışık altında yaklaşık 45 derecelik bir açı ile çekildiği düşünülmektedir.

Dördüncü video görüntüsü; internet ortamından temin edilmiştir, prosesi yapılan madenin niteliği ile ilgili bir bilgi bulunmamaktadır, ışık kullanılmadan doğal olarak yaklaşık 45 derecelik bir açı ile çekildiği düşünülmektedir.

Dört farklı materyal kullanılmasının sebepleri öngörülemeyen hata ve varsayımları minimize etmektir. Böylece farklı madenlerin, farklı flotasyon hücrelerinde, farklı koşullarda, farklı renklerde, farklı ölçeklerde değerlendirilmesi yapılacaktır.

Görüntülerin işlenmesi ve yorumlanması için programlama dili olarak “Python V3” kullanılmıştır. Görüntü işleme kütüphanesi olarak “OpenCV” ve Open CV ye bağlı kütüphane dosyalarından faydalanılmıştır. Tüm bunların çalıştırılması için ise “VS Code” programlama editörü kullanılmıştır. Şekil 2.9’da geliştirme ortamına ait görsel verilmiştir.

Şekil 2.9: VS Code Geliştirme Ortamı



2.4.2. Köpük Hızının Tespiti

Materyaller üzerinde yapılan işlemlerde, her materyal için aynı yöntem uygulanmıştır. Sırasıyla yapılan işlemler şu şekilde özetlenebilir;

Materyallerin orjinal çözünürlükleri ve boyutları üzerinde değişiklik yapılmamıştır, uzantıları “*.mp4” olan video dosyaları olduğu gibi işleme alınmıştır.

Video daki takibi yapılacak köşelerin bulunması için Shi-Tomasi parametreleri ve ayrıca Lucas-Kanade Metodu için kullanılacak parametreler belirlenmiştir.

Görüntü üzerinde takip için kullanılacak noktaların ve akış çizgisinin rengi kırmızı olarak belirlenmiştir.

Görüntüler üzerinde renk desen çevrimi, gri mod uygulaması gibi bir dizi işlem gerçekleştirilmiştir.

Görüntü parçalarının takip işlemi yapılmıştır, akabinde maskeleye yapılarak görüntüler üzerinde kırmızı nokta ve çizgileri eklenmiştir. Şekil 2.10’da bu aşamaların yazılımı ile ilgili görsel verilmiştir.

Şekil 2.10: Köpük Hızının Bulunması, Yazılım Aşaması 1

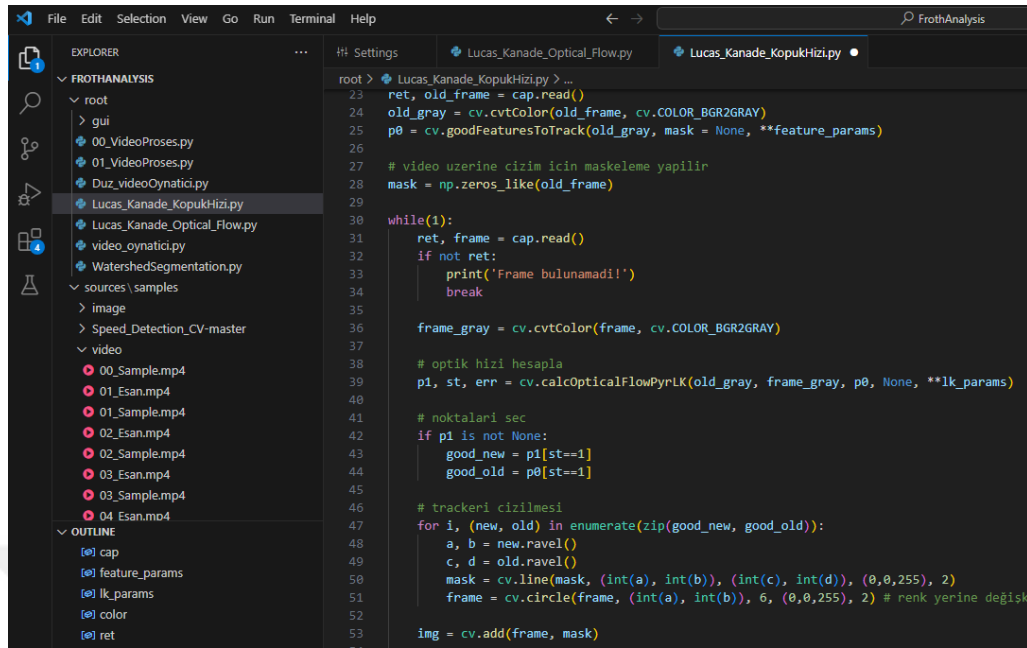
```

1 import numpy as np
2 import cv2 as cv
3 import argparse
4
5
6 cap = cv.VideoCapture('sources/samples/video/02_Sample.mp4') # args.image
7
8 # ShiTomasi ile köşelerin yakalanması
9 feature_params = dict( maxCorners = 100,
10                       qualityLevel = 0.5,
11                       minDistance = 20,
12                       blockSize = 10 )
13
14 # lucas kanade parametreleri
15 lk_params = dict( winSize = (25, 25),
16                 maxLevel = 2,
17                 criteria = (cv.TERM_CRITERIA_EPS | cv.TERM_CRITERIA_COUNT, 10, 0.03))
18
19 # renk oluşturun
20 color = np.random.randint(0, 255, (100, 3))
21
22 # İlk frame i al ve içerisindeki köşeleri bul
23 ret, old_frame = cap.read()
24 old_gray = cv.cvtColor(old_frame, cv.COLOR_BGR2GRAY)
25 p0 = cv.goodFeaturesToTrack(old_gray, mask = None, **feature_params)
26
27 # video üzerine çizim için maskeleye yapilir
28 mask = np.zeros_like(old_frame)

```

Sonrasında, Lucas-Kanade Fonksiyonu çalıştırılarak, video daki her kare için işleme devam edilmiş ve köpük hızı bulunmuştur. Şekil 2.11’de hazırlanan yazılım diğer aşamaları ile ilgili görsel verilmiştir.

Şekil 2.11: Köpük Hızının Bulunması, Yazılım Aşaması 2



```

root > Lucas_Kanade_KopukHizi.py > ...
23 ret, old_frame = cap.read()
24 old_gray = cv.cvtColor(old_frame, cv.COLOR_BGR2GRAY)
25 p0 = cv.goodFeaturesToTrack(old_gray, mask = None, **feature_params)
26
27 # video uzerine cizim icin maskeleme yapilir
28 mask = np.zeros_like(old_frame)
29
30 while(1):
31     ret, frame = cap.read()
32     if not ret:
33         print('Frame bulunamadil!')
34         break
35
36     frame_gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
37
38     # optik hizi hesapla
39     p1, st, err = cv.calcOpticalFlowPyrLK(old_gray, frame_gray, p0, None, **lk_params)
40
41     # noktaları sec
42     if p1 is not None:
43         good_new = p1[st==1]
44         good_old = p0[st==1]
45
46     # trackeri cizilmesi
47     for i, (new, old) in enumerate(zip(good_new, good_old)):
48         a, b = new.ravel()
49         c, d = old.ravel()
50         mask = cv.line(mask, (int(a), int(b)), (int(c), int(d)), (0,0,255), 2)
51         frame = cv.circle(frame, (int(a), int(b)), 6, (0,0,255), 2) # renk yerine deęiştir
52
53     img = cv.add(frame, mask)

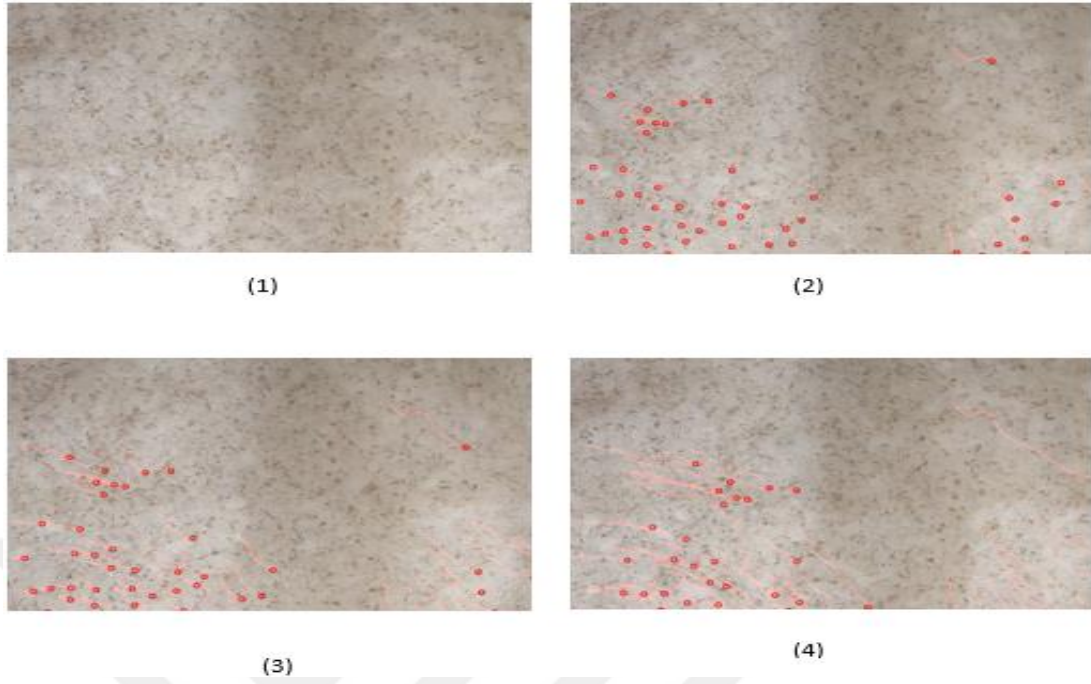
```

Köpük hızının hesaplanması ile ilgili yazılım materyaller kullanılarak çalıştırılmış ve her materyal için köpük akışının hesaplanabildiği görülmüştür.

Videoların sürelerinin farklı olmaları sebebi ile her materyal için aynı anda, aynı sürede örnekleme yapılamamıştır. Bununla birlikte alınan örnekler; (1)- t %0 zamanı, (2)- t %25 zamanı, (3)- t %75 zamanı ve (4)- t %100 zamanı şeklinde her videonun süresi içerisinde 4 örnek alınarak aşağıdaki görsellerde sırasıyla verilmiştir.

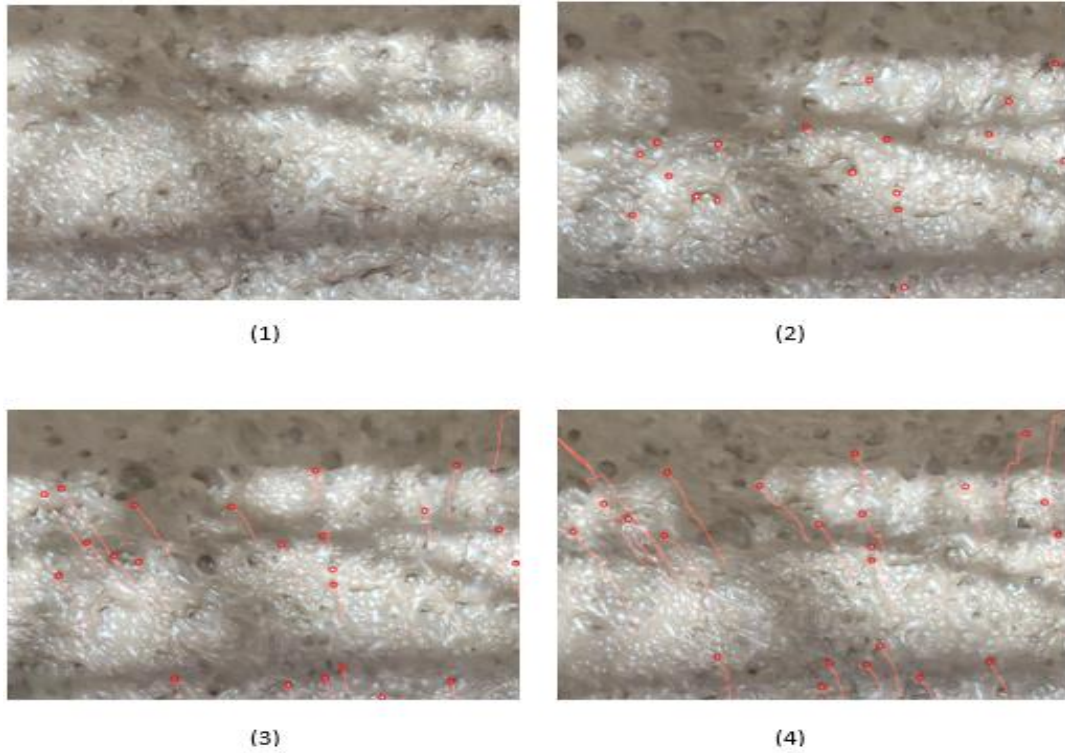
Materyal 1 deki köpük hızına ait ilerleme ve akış görüntüsü Şekil 2.12'de verilmiştir.

Şekil 2.12: Materyal 1, Köpük Hızı ve Akışı



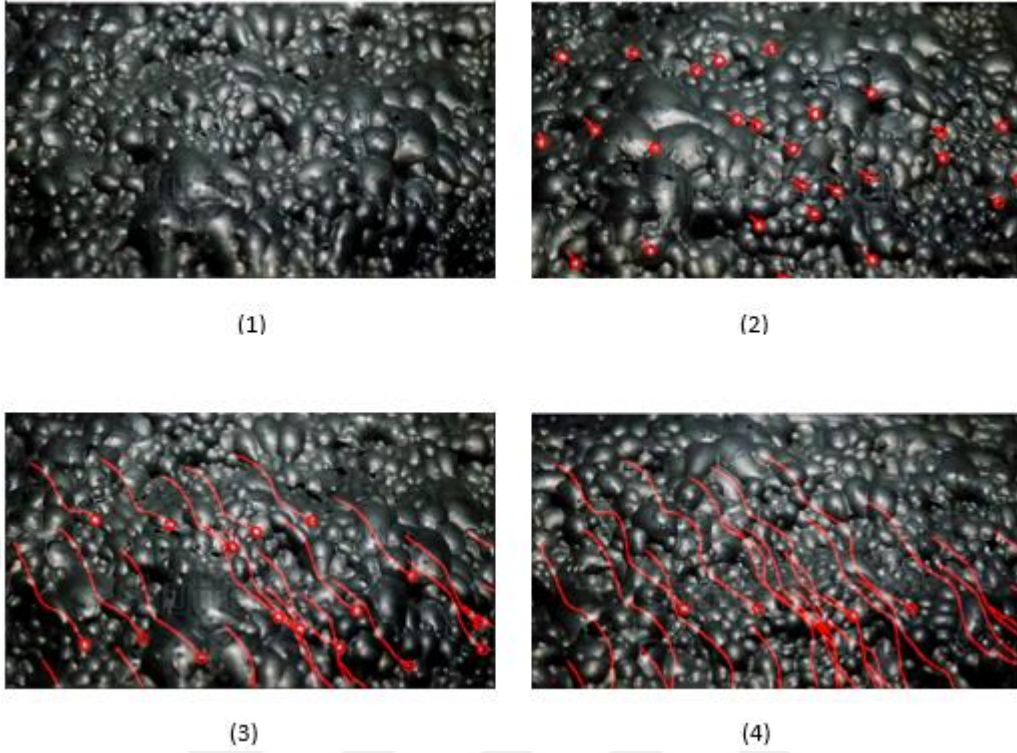
Materyal 2 deki köpük hızına ait ilerleme ve akış görüntüsü Şekil 2.13’de verilmiştir.

Şekil 2.13: Materyal 2, Köpük Hızı ve Akışı



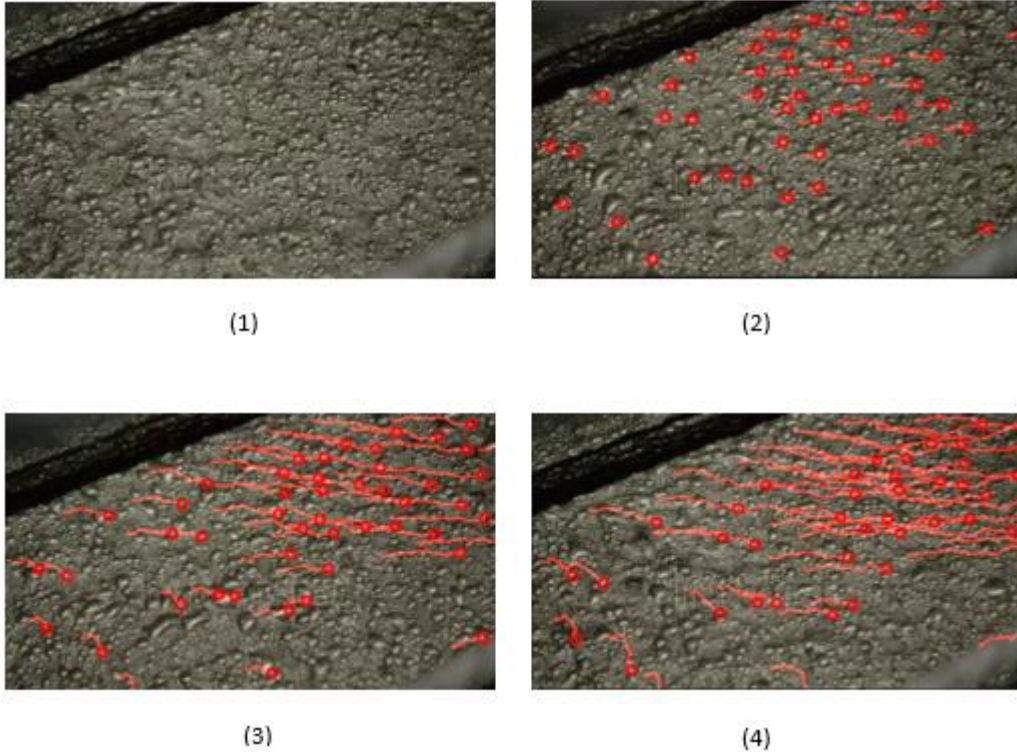
Materyal 3 deki köpük hızına ait ilerleme ve akış görüntüsü Şekil 2.14’de verilmiştir.

Şekil 2.14: Materyal 3, Köpük Hızı ve Akışı



Materyal 4 deki köpük hızına ait ilerleme ve akış görüntüsü Şekil 2.15’de verilmiştir.

Şekil 2.15: Materyal 4, Köpük Hızı ve Akışı

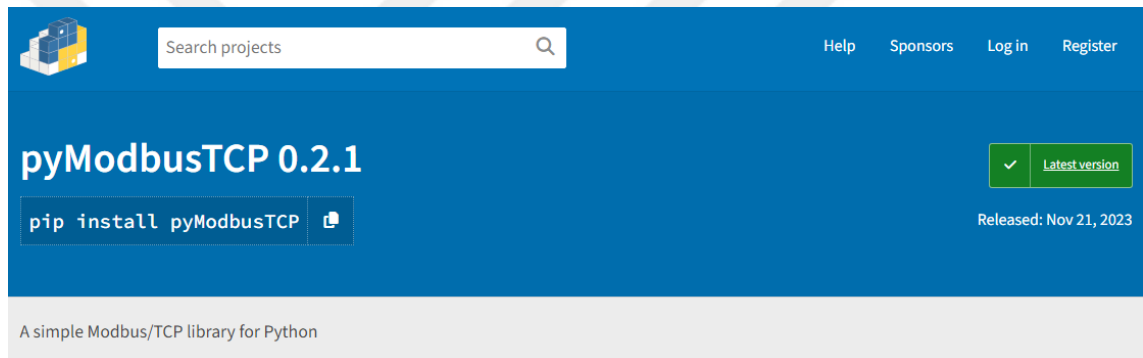


2.4.3. Verinin İletimi

Köpük hızının bulunması sonrası, köpük hızının diğer endüstriyel sistemlerden okunabilmesi gerekmektedir. Bu sebeple endüstriyel otomasyon da yaygın olarak kullanılan ve hemen hemen tüm üreticiler tarafından geliştirici kütüphanelerinde bulunan Modbus TCP protokolü tercih edilmiştir.

Köpük hızı verisinin Modbus TCP protokolü ile PLC ye aktarımı için, görüntü işleme sisteminin Modbus TCP Server olarak çalışması ve PLC nin istemde bulunduğu adresten köpük hızı verisini iletmesi hedeflenmiştir. Bu sebeple görüntü işleme sistemi yazılımında, telif ve lisans hakkı gerektirmeyen, Şekil 2.16’da görseli verilen “pyModbusTCP” kütüphanesi kullanılmıştır.

Şekil 2.16: pyModbusTCP Kütüphanesi



İlgili kütüphane içerisinde bulunan dosyalar, bu çalışmaya uygun olarak düzenlenmiştir. Köpük hızı bilgisi kantitatif bir değişken olarak, modbusTCP protokolü üzerinden veri iletişimi sağlayacaktır. Şekil 2.17’de bu kısımda aktarılan çalışmanın yazılım aşamasına ait bir görsel verilmiştir.

Şekil 2.17: ModbusTCP İletişimi Yazılım Aşaması

```

23
24
25 class ModbusServerRequestHandler(ModbusProtocol):
26     """Implements modbus slave wire protocol.
27
28     This uses the asyncio.Protocol to implement the server protocol.
29
30     When a connection is established, a callback is called.
31     This callback will setup the connection and
32     create and schedule an asyncio.Task and assign it to running_task.
33     """
34
35 def __init__(self, owner):
36     """Initialize."""
37     params = CommParams(
38         comm_name="server",
39         comm_type=owner.comm_params.comm_type,
40         reconnect_delay=0.0,
41         reconnect_delay_max=0.0,
42         timeout_connect=0.0,
43         host=owner.comm_params.source_address[0],
44         port=owner.comm_params.source_address[1],
45     )
46     super().__init__(params, False)
47     self.server = owner
48     self.running = False
49     self.receive_queue = asyncio.Queue()
50     self.handler_task = None # coroutine to be run on asyncio loop
51     self.framer = ModbusFramer
52     self.loop = asyncio.get_running_loop()
53

```

2.4.4. Harici Sistemler

Köpük hızının tespit edilmesi ve diğer sistemler tarafından değerlendirilecek olan iletişim altyapısı tamamlanmıştır. Köpük hızı verisinin harici sistemler ile olan entegrasyonu ve kullanımı ileriki bölümler de detaylı şekilde açıklanacaktır. Bununla birlikte bu sistemin flotasyon tesislerin de endüstriyel sistemlere bağlantısı yapılmadan da kullanılabilir olduğu kanaatine varılmıştır. Endüstriyel bir IP kamera kullanımı ile birlikte, flotasyon operatörleri tarafından sahada bulunan flotasyon hücresinin yanına gitmeden köpük formu izlenebilecektir.



ÜÇÜNCÜ BÖLÜM
BULANIK MANTIK KONTROLÖRÜ

3.1. HÜCRE KONTROLÜNÜN TEMELİ VE ZORLUKLARI

Flotasyon sistemlerinde köpük seviyesinin ve hava debisinin stabil kontrolü, ürün kalitesi ve verimliliği bakımından oldukça önemlidir. Köpük kalitesinin istenilen düzeyde olması için hücrelere blowerlar üzerinden belirlenen debilerde hava gönderilmektedir. Bu sayede köpük istenilen boyuta ulaşır ve savaklardan taşarak ürün kazanımı gerçekleşir.

Klasik kontrol sistemlerinde, köpüğün savaklardan taşması için hücredeki köpük seviyesi göz önüne alınarak otomasyon yazılımında bir seviye set parametre değeri girilir ve hücrenin boşaltma vanaları bu seviyeyi kontrol etmek için belirli oranlarda açıp kapayarak seviyeyi sabit tutmaya çalışır. Aynı şekilde hava debisi de operatörler tarafından belirlenerek hava hattı üzerinde bulunan vananın kontrolü otomasyon yazılımı tarafından yapılmaktadır.

Bu aşamada vanaların, köpük seviyesini ve hava debisini sabit tutması için yaygın olarak PID kontrol metodu kullanılır ve bu metod büyük oranda başarı sağlar; ancak flotasyon sistemlerinin mimarisi üzere her hücrenin çıkış vanasının bir sonraki hücreye boşaltıyor olması, ilk hücreye gelen ürünün debisinin değişken olması, hücrelerin boşaltma vana çıkışlarının zamanla tıkanması, herhangi bir hücredeki dengesizliğin diğer hücreleri de doğrudan etkiliyor olması flotasyon hücresinin kontrolünün zorlukları olup PID kontrol metodunun yetersiz kaldığı durumları oluşturur. Bu bölümde bu handikapların hata oranlarını daha aza indirecek olan çalışma aktarılacaktır.

3.2. TASARIM ORTAMI

Geliştirilecek olan kontrolcü modeli ve yöntemi için, aşağıdaki donanım ve yazılım platformları kullanılmıştır.

Donanım olarak günümüz endüstriyel otomasyon sektöründe ve birçok proses sahasında sıklıkla tercih edilen Siemens marka S7-1513 serisi PLC tercih edilmiştir. Seçilen PLC giriş çıkış modüllerinin esnekliği, haberleşme altyapısının sunduğu destek ile tüm proses kontrol sistemlerinde kullanılabilir düzeyde bir CPU dur.

Kontrolcü içerisinde koşacak kodların geliştirme aşamasında, yazılım simülasyonu kullanılmamış, hazırlanan kodların derlenerek gerçek CPU üzerindeki performansları denenmiştir. Bu sebeple içerisinde PLC ve PLC nin çalışması için gereken

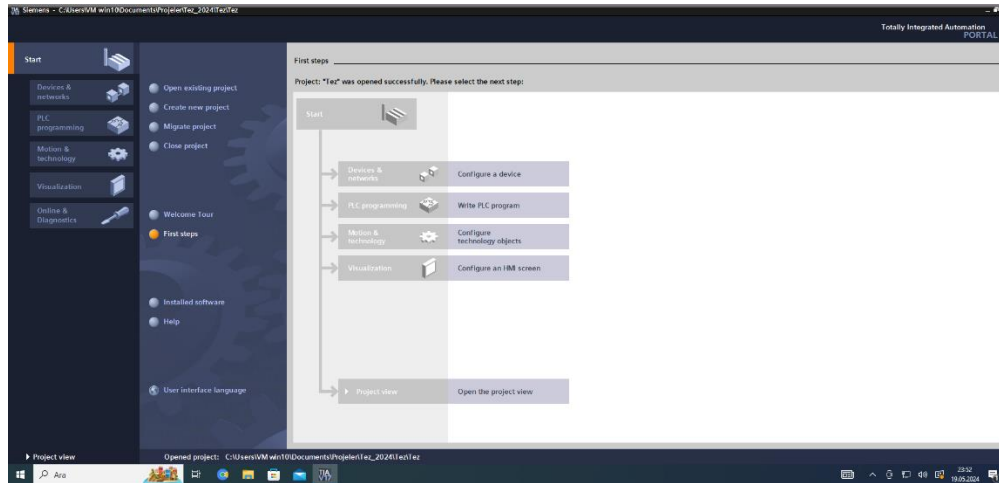
Güç kaynağının bulunduğu masaüstü bir deney seti hazırlanmıştır. Görsel 3.1’de yazılım geliştirme ortamı için kullanılan deney setinin görseli verilmiştir.

Görsel 3.1: S7-1513 PLC Masaüstü Deney Seti



PLC yazılımı geliştirme ortamı, yine Siemens ürün gruplarının programlanması ve konfigürasyonları için sunulan TIA Portal v17 yazılımı ile yapılmıştır. TIA (Totally Integrated Automation) Portal gelişmiş özellikleri, IEC 61131 standartlarında farklı kodlama dillerini destekleyen, içerisinde bulundurduğu kütüphanelerin zenginliği ile yazılım geliştiriciler için oldukça güçlü bir arayüz sunmaktadır. Şekil 3.1’de TIA Portal arayüzü ile ilgili bir görsel verilmiştir.

Şekil 3.1: TIA Portal Yazılım Geliştirme Ortamı



Geliştirilen yazılımın her ne kadar gerçek CPU üzerinde koşturulması sağlansa da gerçek zamanlı olarak bir tesiste denenmesi bu aşamada mümkün olmamıştır. Bu sebeple tesisi simüle ederek gerçek zamanlı tepkilerin alınması ve yazılımın bu geri

dönüŖlere istinaden sistemi hangi düzeyde, hangi dođrulukta kontrol edebileceđi Factory IO isimli simülasyon programı konfigüre edilerek yapılmıŖtır.

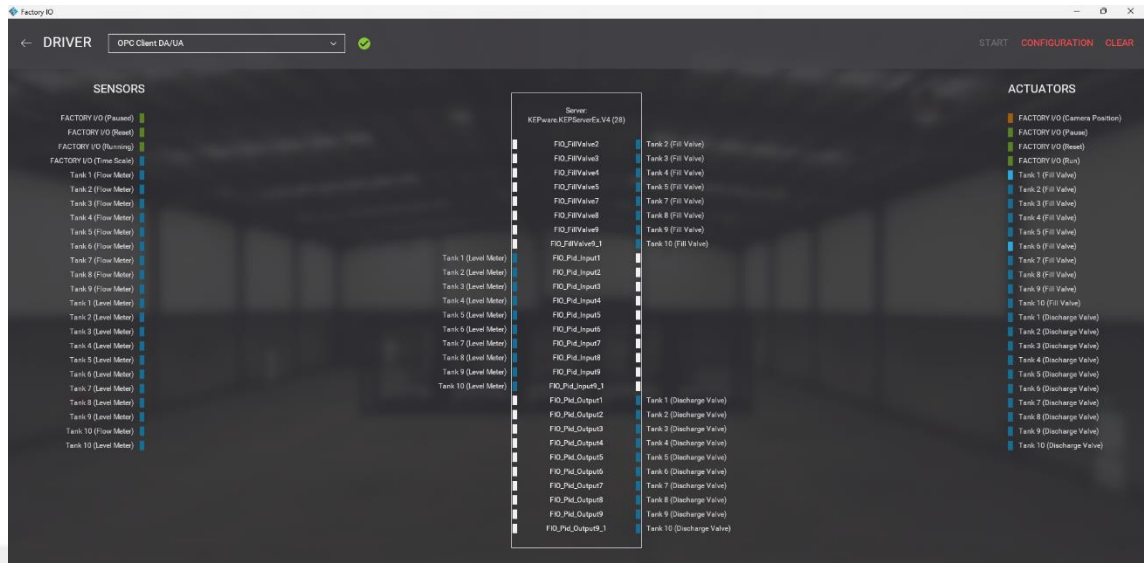
Factory IO yazılım kullanılarak, toplamda 2 flotasyon sistemi bankı ve her bankta 5 flotasyon hücresi bulunan 10 hücreli bir flotasyon sistemi modellenmiŖtir. Her flotasyon hücresi için enstrümantasyon dizaynı yapılmıŖ ve gerçek bir tesisteki gibi modellenmiŖtir. Görsel 3.2’de flotasyon hücresinin/hücrelerinin enstrümantasyonuna ait bir görsel verilmiŖtir.

Görsel 3.2: Factory IO, Enstrümantasyon Modellemesi



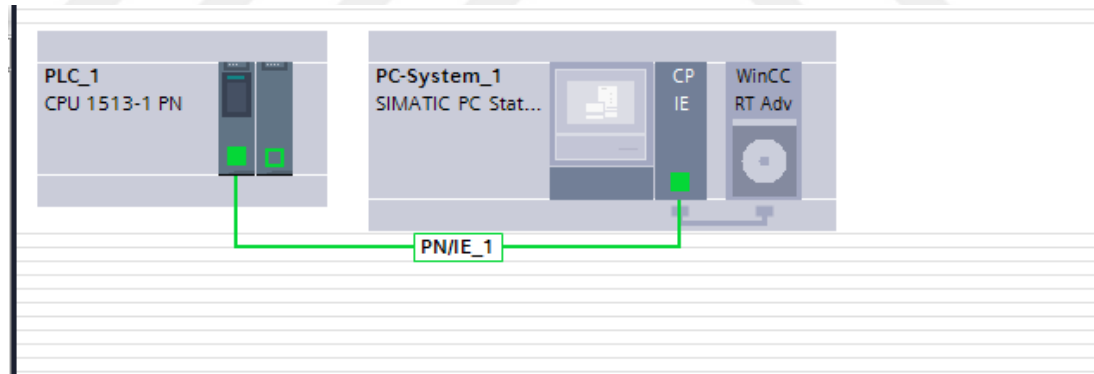
Factory IO sunduđu imkanlar ile endüstriyel yazılım geliŖtiriciler için yeterli düzeyde olanak sunmaktadır. Tesis simülasyonun gerçek zamanlı olarak yapılabilmesi ve PLC içerisinde çalıŖan kodların gerçek zamanlı analiz edilebilmesi için, endüstriyel kontrolcüler ile OPC (Ole Process Control) server üzerinden iletişim kurabilmektedir. Böylece simülasyon için modellenen enstrümanlardan alınan/gönderilen veriler gerçek bir tesistekinden farklı olmayacak kadar efektif olarak kullanılabilir. Ŗekil 3.2’de Factory IO, OPC Server konfigürasyonu ile ilgili bir görsel verilmiŖtir.

Şekil 3.2: Factory IO OPC Server Konfigürasyonu



Siemens S7-1513 serisi PLC nin, TIA Portal üzerindeki konfigürasyonu, WinCC RT (Real Time) eklentisi ile Factory IO yazılımı ile eşleştirilmiştir. İki sistem arasındaki veri transferi OPC server üzerinden yapılmıştır. Bu konfigürasyon ile ilgili görsel Şekil 3.3'de gösterilmiştir.

Şekil 3.3: TIA Portal Simülasyon Konfigürasyonu



Derlenen yazılımın PLC içerisine yüklenmesi ve çalıştırılması ile birlikte, Grösel 3.3'de görseli verilen Factory IO programındaki tesis simülasyonu çalıştırılarak, hazırlanan yazılımın hangi düzeyde çalıştığı gözlemlenmiş ve gerekli revizyonlar yapılarak yazılım tamamlanmıştır.

Görsel 3.3: Flotasyon Tesisi Simülasyon Arayüzü



3.3. KONTROLÖR TASARIMI

Çok yaygın olarak kullanılan PID kontrol algoritmasının efektif olarak çalışması ile birlikte, daha önce aktarılan sebepler dolayısı ile geliştirilmesi yönünde bir çalışma yapılmıştır. Bu çalışma temel olarak PID kontrol bloğunun çıkışındaki kontrol değişkeninin manipüle edilmesine dayanmaktadır. PID kontrol bloğu hemen hemen tüm endüstriyel yazılım uygulamalarında, üreticiler tarafından standart kütüphanelerde bulunmaktadır. Bu sebeple yapılan çalışmada TIA Portal kütüphaneleri tarafından sunulan standart PID bloğu kullanılmıştır. PID kontrol bloğunun çıkışındaki değişken Bulanık Mantık temelli bir tasarım ile desteklenmiş, böylece tümleşik ve daha güçlü bir kontrol sistemi hedeflenmiştir. Hedeflenen model geliştirilir iken sadece flotasyon hücrelerindeki kontroller için değil genel olarak tüm proses kontrol uygulamalarında kullanılmasının mümkün olabilmesi için FB (Function Block) olarak dizayn edilmiştir. Genel olarak bulanık mantık temelli kontrol bloklarının endüstriyel yazılım kütüphanelerinde bulunmadığı söylenebilir.

3.3.1. Bulanık Mantık Yaklaşımı

Bulanık terimi, belirsiz veya belirsiz olan şeyleri ifade eder. Gerçek dünyada birçok kez, bir durumun doğru mu yoksa yanlış mı olduğunu belirlenemez, burada bulanık mantık, akıl yürütme için değerli bir esneklik sağlar. Bu şekilde, herhangi bir durumun belirsizliklerini ve belirsizlikleri göz önünde bulundurulabilir.

Boolean sistemde doğruluk değeri, “1” kesin doğruluk değerini temsil ederken, “0” kesin yanlış değerini temsil eder, ancak bulanık sistemde, değişkenlerin doğruluk değerleri hem tamamen doğru hem de tamamen yanlış arasında değişen herhangi bir gerçek sayı olabilir. Bu kısmi doğruluk kavramını ele almak için kullanılır, burada doğruluk değeri kesin doğru ve kesin yanlış arasında değişebilir.

Bulanıklaştırma modeli; numerik girişleri belirli bir üyelik derecesine sahip bulanık kümelerle eşleştirme sürecidir. Eşleştirme bir tablo oluşturularak yapılır. Bu tablo uzmanlarca proses gözlemi yapılarak ve istatistiksel analizler sonucu oluşturulur ve kontrol algoritması bu tabloya göre kodlanır. Bu tablo Boolean sistemindeki kesin doğru “1”, kesin yanlış “0” değerlerinden ziyade; kesin doğru, neredeyse doğru, ne doğru ne yanlış, neredeyse yanlış, kesin yanlış gibi ifadelerle doldurulur. Bu çalışmada üyelik fonksiyonları hata ve hata değişimine göre şu şekilde planlanmıştır:

NL : Negatif yönde fazla,

NM : Negatif yönde orta,

NS : Negatif yönde az,

ZE : Sıfır,

PS : Pozitif yönde az,

PM : Pozitif yönde orta,

PL : Pozitif yönde fazla,

Buradaki üyelik fonksiyonlarına göre hata tespiti yapılacak olup, yapılan çalışmada klasik PID kontrol metodunu denetleme ve müdahale etme görevini üstlenecektir.

3.3.2. Kural Tablosu

Uygulanan metotta PID kontrol metodunun tepkilerini hızlandırmak ve güvenli seviyelerde tutmak için PID bloğunun çıkışını kontrol eden ek bir yazılım hedeflenmiştir. PID kontrol sistemi her ne kadar güçlü bir kontrol algoritması olsa da kontrol mühendisinin tecrübesi ve PID parametrelerini doğru ayarlaması ile mümkün olmaktadır. Bu çalışma PID bloğunun aşırı salınım durumlarını engellemek ve öngörülen parametrelerin daha efektif çalışmasını sağlamak amacıyla hazırlanmıştır.

Bir tablo oluşturularak hata ve hatadaki değişimler negatif ve pozitif yönde; az, orta ve çok şeklinde sınıflandırılmış olup buradan hata tespiti yapıldıktan sonra çıkışa direkt olarak etki etmektedir. Tablo 3.1 bu çalışma için hazırlanan kural tablosunu göstermektedir.

Tablo 3.1: Bulanıklaştırma Kural Tablosu

	1	2	3	4	5	6	7	8
DURUM	$E \leq NL$	$E > NL \ \& \ E \leq NM$	$E > NM \ \& \ E \leq NS$	$E > NS \ \& \ E \leq 0$	$E > 0 \ \& \ E \leq PS$	$E > PS \ \& \ E \leq PM$	$E > PM \ \& \ E \leq PL$	$E > PL$
1 $ER \leq NL$	NL	NM	NM	NM	NM	NS	NS	PM
2 $ER > NL \ \& \ ER \leq NM$	NL	NM	NM	NS	ZE	ZE	ZE	PM
3 $ER > NM \ \& \ ER \leq NS$	NL	NM	NS	ZE	ZE	ZE	ZE	PM
4 $ER > NS \ \& \ ER \leq 0$	NL	NM	ZE	ZE	ZE	ZE	PS	PL
5 $ER > 0 \ \& \ ER \leq PS$	NL	NM	ZE	ZE	ZE	ZE	PM	PL
6 $ER > PS \ \& \ ER \leq PM$	NL	NM	ZE	ZE	ZE	ZE	PM	PL
7 $ER > PM \ \& \ ER \leq PL$	NL	NM	NS	ZE	ZE	PS	PL	PL
8 $ER > PL$	NM	NS	NS	PM	PM	PM	PL	PL

Kural tablosunda yatay bölümde E (error) yani hata, dikey bölümde ise ER (error rate) sıralanmıştır. PLC nin her çevrim süresinde kontrol sistemindeki mevcut hatanın, bir önceki çevrim döngüsündeki farkı ile hata oranı belirlenmiştir. Sonrasında E ve ER arasındaki kurallar belirlenerek kategorize edilmiştir.

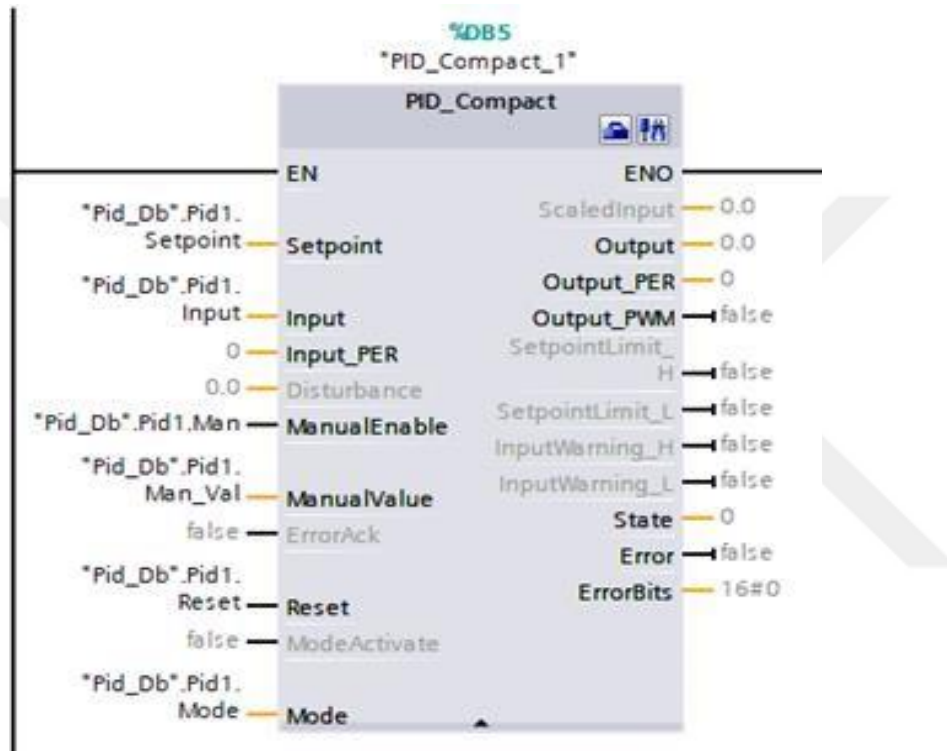
3.3.3. Algoritma

Algoritma hazırlanırken öncelikli olarak TIA Portal kütüphanesinde bulunan ve Şekil 3.4’de verilen standart PID bloğu kullanılmıştır. Blok girişine kontrol edilecek değişkenin istenilen değeri ve proses sonucu elde edilen değeri girilmiştir. PID bloğu çıkışındaki kontrol değişkeni ise bulanık mantık temelli hazırlanan görseli Şekil 3.8’de verilen PID_Regulator bloğun girişine bağlanmıştır. Bu algoritma istenen proses değeri (set değeri) ve mevcut proses değeri arasındaki farkı ve değişim oranını, girilen zamana göre analiz ederek oluşturulan kural tablosuna göre çıkışa girilen hata oranlarına göre etki eder.

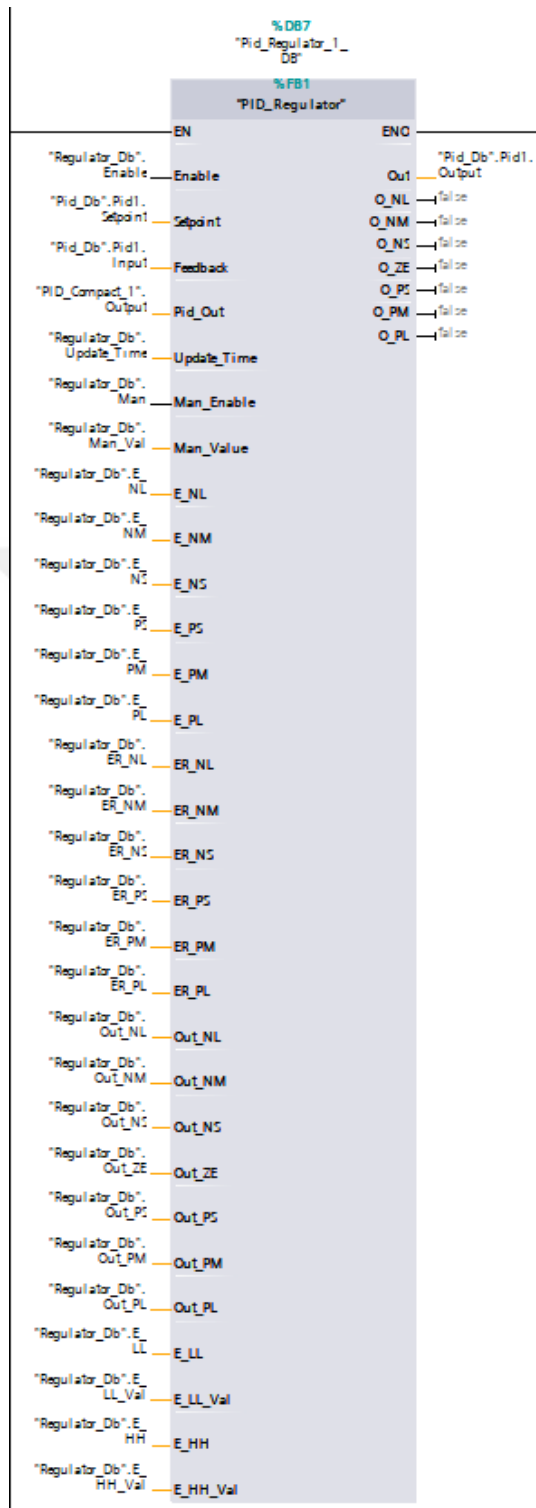
Bloğun ekstra iki kritik limite de vardır, böylece kritik limitlerin dışına çıktığı durumlarda PID ve hataya bağımlı düzeltme işlemlerinden bağımsız olarak, çıkış enstrümanlarına istenilen oranda set değeri gönderir. Bloğun amacı PID salınımlarını

minimumuna indirmek ve istenilen proses değerine daha çabuk sürede ulaşmasını sağlamaktır. Kural tablosunda ZE ile gösterilmiş alanlarda direkt olarak PID çıkışını, bulanık mantık temelli PID regülatör çıkışına aktarır ve kontrolü PID kontrolcüsü sağlar. Çıkıştaki O_NL, O_NM, O_NS, O_ZE, O_PS, O_PM, O_PL bitleri anlık hangi yaklaşımın uygulandığını gösterir, buradan proses takibi yapılabilir ve giriş parametreleri optimize edilebilir.

Şekil 3.4: TIA Portal Standart PID bloğu



Şekil 3.5: Bulanık Mantık Temelli PID_Regulator Bloğu



3.3.4. Yazılım

Kodlama, IEC 61131-3 standartlarına uygun olarak ST (Structure Text) [IEC 61131-3, 2013] dili ile geliştirilmiştir. ST eskiden yaygın olarak kullanılan Pascal dili ile çok benzer bir geliştirme ortamıdır. Kodun içerisindeki tanımlamalar açıklamaları ile

verilmiştir. Kodlama yapılır iken programlama editörlerinin Türkçe karakterleri desteklememesi sebebi ile tüm değişkenler İngilizce olarak tanımlanmıştır. Bununla birlikte açıklama satırları derlenmediği için bu bölümlerde türkçe açıklamalara yer verilmiştir. Yapılan çalışma genel bir kontrol metodu olarak tasarlanmış olup PID döngüsü ile kontrol edilebilecek tüm nicelikler için uygulanabilecektir.

Şekil 3.6'da geliştirilen yazılımda tanımlanan değişkenlerin bulunduğu bölümün görseli verilmiştir. Tablo 3.1'de verilen değişkenlerin PLC deki tanımlamaları bu kısımda yer almaktadır.

Şekil 3.6: PID_Regulator Değişken Listesi

Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision	Comment
Static									
Enable	Bool	true	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Update_Time	UInt	500	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Man	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Man_Val	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
E_NL	Real	-3.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
E_NM	Real	-1.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
E_NS	Real	-0.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
E_PS	Real	0.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
E_PM	Real	1.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
E_PL	Real	3.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
ER_NL	Real	-1.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
ER_NM	Real	-0.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
ER_NS	Real	-0.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
ER_PS	Real	0.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
ER_PM	Real	0.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
ER_PL	Real	1.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Out_NL	Real	-10.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Out_NM	Real	-7.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Out_NS	Real	-2.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Out_ZE	Real	0.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Out_PS	Real	2.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Out_PM	Real	7.5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Out_PL	Real	10.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
E_LL	Real	-4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
E_LL_Val	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
E_HH	Real	4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
E_HH_Val	Real	100.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Değişkenlerin tanımlanması sonrası zaman sabitlerinin belirlenmesi ve proses hatasının belirlenmesi hazırlanmıştır. Şekil 3.7'de yazılımın bu kısmı ile ilgili görsel verilmiştir.

Şekil 3.7: PID_Regulator Yazılım Aşaması 1

```

S7_1500 > PLC_1 [CPU 1513-1 PN] > Program blocks > Pid > PID_Regulator [FB1]
Block interface
IF... CASE... FOR... WHILE... (*..*) REGION
OF... TO DO... DO...
1
2
3 // Update Time Limits
4 IF #Update_Time < 10 THEN
5   #Tl_Time := T#10ms;
6 ELSIF #Update_Time > 10000 THEN
7   #Tl_Time := T#10000ms;
8 ELSE
9   #Tl_Time := UINT_TO_TIME(#Update_Time);
10 END_IF;
11
12
13 // Error and Error Rate Calculations
14
15 #Rl(CLK := #Tl.Q);
16
17 IF #Rl.Q THEN
18   #Process_Error := #Feedback - #Setpoint;
19   #Error_Rate_First := #Error_Rate_Last;
20   #Error_Rate_Last := #Feedback;
21   #Process_Error_Rate := #Error_Rate_Last - #Error_Rate_First;
22 END_IF;
23
24 // Update Timer
25 #Tl_TON(IN := NOT #Tl.Q,
26         PT := #Tl_Time);
27
28 // E <= NL
29 IF #Process_Error <= #E_NL THEN
30
31   // ER <= NL
32   IF #Process_Error_Rate <= #ER_NL THEN

```

Belirlenen kurallara göre oluşturulan kodlar ise yazılımın ilerleyen aşamalarında yer almıştır. Şekil 3.8’de kurallara göre yazılan kod satırlarına ait görsel verilmiştir.

Şekil 3.8: PID_Regulator Yazılım Aşaması 2

```

S7_1500 > PLC_1 [CPU 1513-1 PN] > Program blocks > Pid > PID_Regulator [FB1]
Block interface
IF... CASE... FOR... WHILE... (*..*) REGION
OF... TO DO... DO...
242
243 // E > NS & E <= 0
244 IF #Process_Error > #E_NS AND #Process_Error <= 0 THEN
245
246   // ER <= NL
247   IF #Process_Error_Rate <= #ER_NL THEN
248     #States[4, 1] := TRUE;
249   ELSE
250     #States[4, 1] := FALSE;
251   END_IF;
252
253   // ER > NL & ER <= NM
254   IF #Process_Error_Rate > #ER_NL AND #Process_Error_Rate <= #ER_NM THEN
255     #States[4, 2] := TRUE;
256   ELSE
257     #States[4, 2] := FALSE;
258   END_IF;
259
260   // ER > NM & ER <= NS
261   IF #Process_Error_Rate > #ER_NM AND #Process_Error_Rate <= #ER_NS THEN
262     #States[4, 3] := TRUE;
263   ELSE
264     #States[4, 3] := FALSE;
265   END_IF;
266
267   // ER > NS & ER <= 0
268   IF #Process_Error_Rate > #ER_NS AND #Process_Error_Rate <= 0 THEN
269     #States[4, 4] := TRUE;
270   ELSE
271     #States[4, 4] := FALSE;
272   END_IF;
273

```

Zamana bağılı yapılan kontroller ve kurallar neticesinde blok çıkışları hazırlanır, Şekil 3.9'da yazılımın bu aşaması ile ilgili görsel verilmiştir.

Şekil 3.9: PID_Regülatör Yazılım Aşaması 3

```

669 /// Out Regulating
670
671 IF #Enable THEN
672
673 IF #Man_Enable THEN
674 #Out := #Pid_Out + #Man_Value;
675 ELSIF #E_LL_En THEN
676 #Out := #E_LL_Val;
677 ELSIF #E_HH_En THEN
678 #Out := #E_HH_Val;
679 ELSE
680 IF #State_NL THEN
681 #Out := LIMIT(MN := 0.0, IN := #Pid_Out + #Out_NL, MX := 100.0);
682 ELSIF #State_NM THEN
683 #Out := LIMIT(MN := 0.0, IN := #Pid_Out + #Out_NM, MX := 100.0);
684 ELSIF #State_NS THEN
685 #Out := LIMIT(MN := 0.0, IN := #Pid_Out + #Out_NS, MX := 100.0);
686 ELSIF #State_ZE THEN
687 #Out := LIMIT(MN := 0.0, IN := #Pid_Out + #Out_ZE, MX := 100.0);
688 ELSIF #State_PS THEN
689 #Out := LIMIT(MN := 0.0, IN := #Pid_Out + #Out_PS, MX := 100.0);
690 ELSIF #State_PM THEN
691 #Out := LIMIT(MN := 0.0, IN := #Pid_Out + #Out_PM, MX := 100.0);
692 ELSIF #State_PL THEN
693 #Out := LIMIT(MN := 0.0, IN := #Pid_Out + #Out_PL, MX := 100.0);
694 END_IF;
695 END_IF;
696
697
698 ELSE
699 #Out := #Pid_Out;
700 END_IF;

```

3.4. SIMÜLASYON

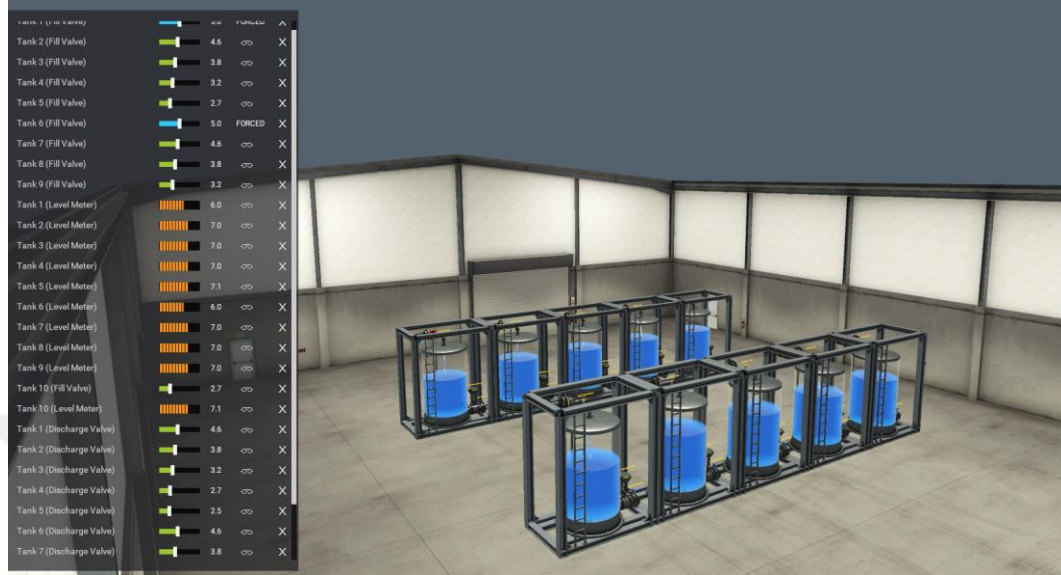
3.4.1. Factory IO Tasarımı

Factory IO Programının içerisinde bulundurduğu zengin kütüphanesi ile sanal bir fabrika oluşturmayı sağlayan bir simülasyon programı olduğuna kısaca değinilmişti. Bu program kullanılarak yapılan sanal bir flotasyon tesisi modeli Şekil 3.10'da gösterilmiştir. Bu model kullanılarak ve PLC içerisinde derlenen program koşturularak sistemin simülasyonu gerçekleştirilmiştir.

Endüstriyel bir PLC ile gerçek zamanlı testler yapılmasına imkân sunduğundan bu program tercih edilmiştir. Kütüphanesinde bulunan tanklar, bu tanklara entegre edilen enstrümantasyon ürünleri (sensörler ve vanalar) analog olarak kontrol edilebilmektedir. Tüm analog veriler 0-10 Volt aralığındadır ve PLC yazılımında % 0-100 olarak anlamlandırılmıştır. Factory IO ve uygulamada kullanılan S7-1513 PLC, OPC server

üzerinden endüstriyel haberleşme protokollerinden biri olan Modbus TCP formatı ile haberleştirilmiştir.

Şekil 3.10: Factory IO Flotasyon Tesis Modeli



Simülasyonun gerçeğe en yakın sonuçlar verebilmesi bakımından her flotasyon hücresinin çıkış vanasının açıklık oranı aynı zamanda bir sonraki hücrenin doldurma vanasının açıklık oranına eşittir. Bu eşitleme işlemleri PLC yazılımında gerçekleştirilmiştir.

Toplamda 10 adet flotasyon hücresi vardır. İlk 5 hücre bu çalışma kapsamında hazırlanan, bulanık mantık temelli regülatör bloğu ile kontrol edilmiştir, diğer 5 hücre ise klasik PID yöntemiyle proses kontrolü yapacaktır.

Anlık olarak aynı anda iki ayrı sisteme de aynı proses set değeri verilmiş ve değerler kaydedilmiştir. Bu değerler grafik anlatım üzerinden kıyaslanmış ve karşılaştırılmaları yapılmıştır.

3.4.2. Klasik PID Kontrolü

PID Kontrolü çok farklı sistemlerde uygulandığı gibi, her sistemde sisteme uygun katsayılara ulaşmak çoğunlukla uğraştırıcı bir süreç almaktadır ve kontrol mühendisinin tecrübesi önemli hale gelmektedir. Sistemi tanımak, sistem içerisinde kullanılan enstrümanların tepkisini anlamak bu süreç için oldukça önemlidir. Devreye alma ve test süreçlerinde sistem farklı senaryolara göre test edilerek bu işlemler gerçekleştirilir ve en uygun olduğu gözlemlenen katsayılarla devreye alma gerçekleşir.

Katsayıların açıklamaları şu şekildedir:

P : Proportional (Kazanç) Katsayısı

I : Integral (İntegral) Katsayısı

D : Derivative (Türev) Katsayısı

ve aşağıdaki formül kapsamında bir değer üretir

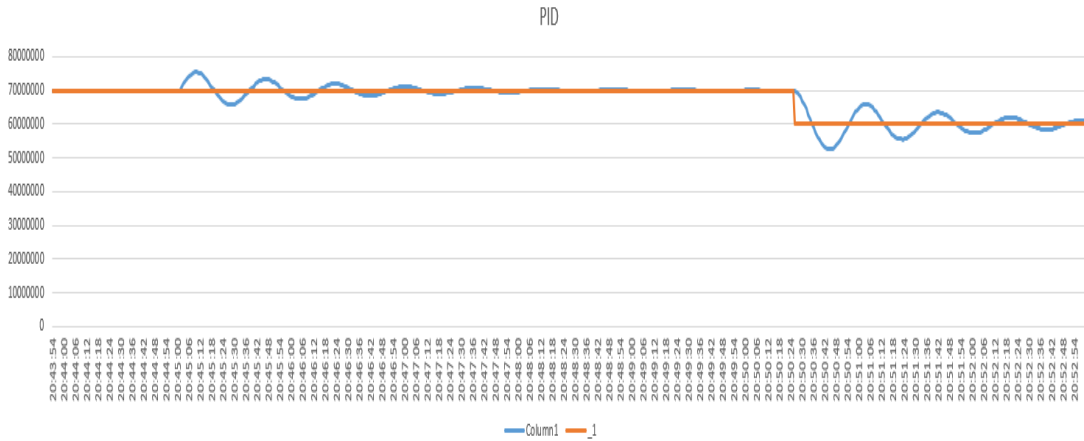
$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} e(t) \right) \quad (3.1)$$

Bu parametreler PID bloğunun temellerini oluştursa da, PID kontrolünün hata tespiti için belirlenen güncelleme zamanı da oldukça önemlidir. Güncelleme zamanı en başta girilmesi gereken parametre olup P, I, D katsayıları mevcut güncelleme zamanına göre ayarlanmalıdır. Aksi takdirde hata tespit algoritmalarındaki sayılar değişeceğinden P, I, D katsayılarının da değişmesi gerekir. Bu uygulamada güncelleme zamanı hızlı tepki verebilmesi için 100 ms olarak belirlenmiştir.

Katsayı belirleme işlemlerinden önce Factory IO programındaki hücreler manuel olarak doldurularak ve boşaltılarak gözlemler gerçekleştirilmiştir. Daha sonra bu alanda basit bir yaklaşım metodu olan Ziegler Nichols yaklaşımı kullanılarak P=1, I=1, D=0,1 olarak ayarlanıp teste başlanmıştır. Ziegler-Nichols tasarım yöntemi, bir PID denetleyicisinin parametrelerini belirlemek için proses kontrolde kullanılan yöntemlerden biridir [Zeynep Y.H.,2023]. Sürekli salınımlara dayanan bir deneme yanılma yöntemidir. Sürekli döngü yöntemi olarak da bilinir [Babu ve Swarnalath, 2017].

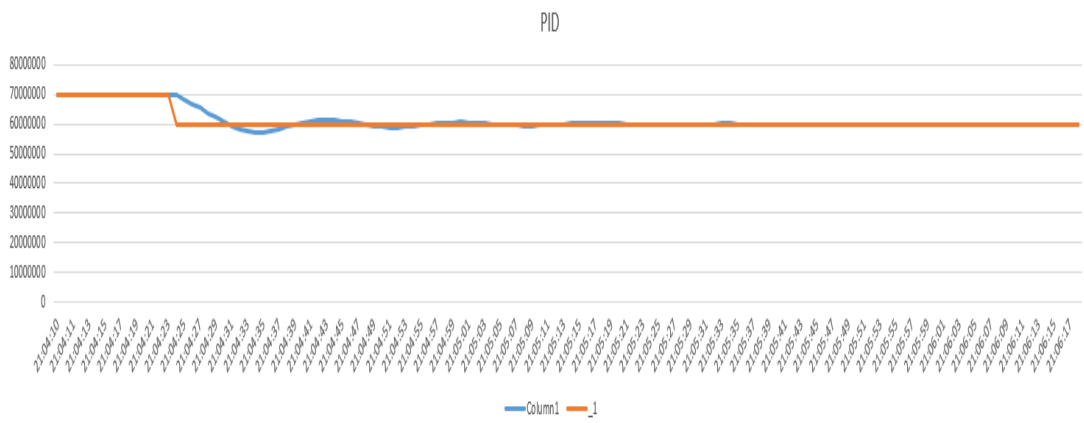
Bu parametrelerle yapılan testte salınımların fazla olduğu saptanmıştır. İlgili parametrelendirme sonrası, proses değişkeninin gösterdiği tepki Şekil 3.11'de gösterilmiştir.

Şekil 3.11: PID Parametreleri (P:1, I:1, D:0) Kontrol Salınımı



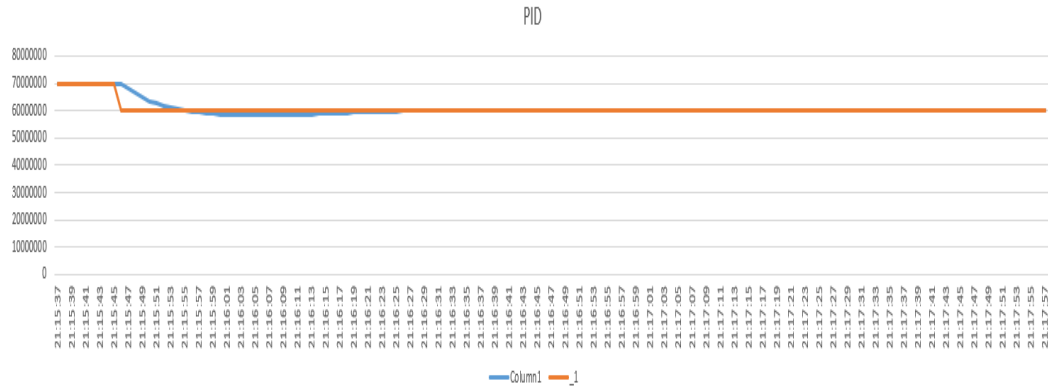
Ardından P katsayısı I katsayısı ve D katsayısı kademe kademe artırılarak testlere devam edilmiştir, P= 5, I=2, D=0,1 parametreleriyle yapılan testin sonuçları Şekil 3.12’de gösterilmiştir.

Şekil 3.12: PID Parametreleri (P:5, I:2, D:0,1) Kontrol Salınımı



Mevcut parametrelerle sistem neredeyse ayarlanmış olup bu aşamadan sonra P ve D katsayısı sabitlenmiştir, I katsayısı artırılarak testlere devam edilmiştir. P= 5, I=10, D=0,1 parametreleriyle yapılan testin grafiği Şekil 3.13’de gösterilmiştir.

Şekil 3.13: PID Parametreleri (P:5, I:10, D:0,1) Kontrol Salınımı



Bundan sonraki yapılacak kıyaslamalı testlerde bu katsayılar kullanılacak ve her flotasyon hücresinde bu parametreler ile devam edilecektir.

3.4.3. PID Regülatör Katsayıları

Bulanık mantık temeli ile hazırlanan Regülatör bloğu PID kontrolcüsüyle birlikte çalışacağı için P, I, D katsayılarını ayarlarken yapılan testler regülatörün parametrelerini ayarlamak için de yapılmıştır. Güncelleme zamanı “Update_Time”, bu blokta 500 ms olarak belirlenmiştir. PID güncelleme zamanından fazla olmasının sebebi bloğun PID’ den bağımsız salınım yaratmaması içindir, diğer parametrelerin ayarlanmasında güncelleme zamanı parametresinin yine oldukça önemi vardır.

Bloğun giriş parametreleri E ve ER kısaltmalarıyla E (Hata), ER (Hata değişimi) kodlanmıştır. E ve ER parametreleri; NL (Negatif yönde aşırı), NM (Negatif yönde orta), NS (Negatif yönde az), PS (Pozitif yönde az), PM (Pozitif yönde orta), PL (Pozitif yönde aşırı) olarak sınıflandırılmıştır.

Sistemin tepkisi, vanaların açma kapama hızı, flotasyon hücresinin dolma hızı, istenilen seviye ve seviye kontrol hassasiyetleri farklı testlere tabi tutularak farklı değerlerle gözlemlenmiştir.

Modellenen sistemdeki flotasyon hücrelerinin giriş ve çıkış debilerinin aynı olduğu varsayılarak, E ve ER değerleri negatif ve pozitif yönlerde simetrik olmasına karar verilmiştir. Yani NS = PS, NM = PM, NL = PL şeklinde olmuştur.

Bloğun hata değerinin kritik limitlerin dışına çıkmasını engelleyen iki parametresi vardır, bunlar: E_LL (Düşük kritik limit) ve E_HH (Yüksek kritik limit) dir. Bu parametreler -4 ve +4 olarak girilmiştir. Bu fonksiyon aktif olduğunda PID ve

Regülatör çıkışından bağımsız E_LL_Val (Düşük kritik limit çıkış değeri) ve E_HH_Val (Yüksek kritik limit çıkış değeri) değerlerini direkt olarak çıkışa yansıtır. Bu uygulamada E_LL_Val = 0, E_HH_Val = 100 olarak girilmiştir. Seviye Değeri - 4 < Set Değeri durumunda vana set değeri 0 , Seviye Değeri > Set Değeri + 4 durumunda vana set değeri 100 olacaktır. Bu iki parametreye göre E_NL, E_NM, E_NS, E_PS, E_PM, E_PL parametrelerinin fonksiyonel olabilmesi için -4 ve +4 arasında olmalıdır ve uygulamada değerler bu aralıkta testlerle seçilmiştir.

Hata değişim parametreleri “ER” sistemdeki salınımları azaltan önemli parametrelerdir. Anlık seviye değişimlerinde yahut seviye set değeri değişimlerinde PID salınımları bariz miktarda artış gösterir, düzenli akışın olduğu sistemlerde belirli bir süre sonrasında sistem kendini toplayacaktır ancak akış düzensiz ise hücreler aşırı boşalmaya ya da aşırı dolmaya devam eder. Düzenli akışın olduğu sistemlerde bile harici sebeplerden kaynaklanan problemler salınımın yol açar ve bu salınım flotasyon hücresinden diğer flotasyon hücresine genlikleri artarak devam eder. ER parametreleri bu soruna çözüm oluşturur. Hata artış oranını takip ederek PID tepki gecikmesinden kaynaklı problemleri minimize ederek sistemin daha az salınım maruz kalmasını sağlar. Bu parametreler yine hata parametrelerinde olduğu gibi kademeli sınıflandırılmıştır. Flotasyon hücresinin dolma hızı ve boşalma hızına göre farklı değerler girilerek optimize edilmeye çalışılmıştır.

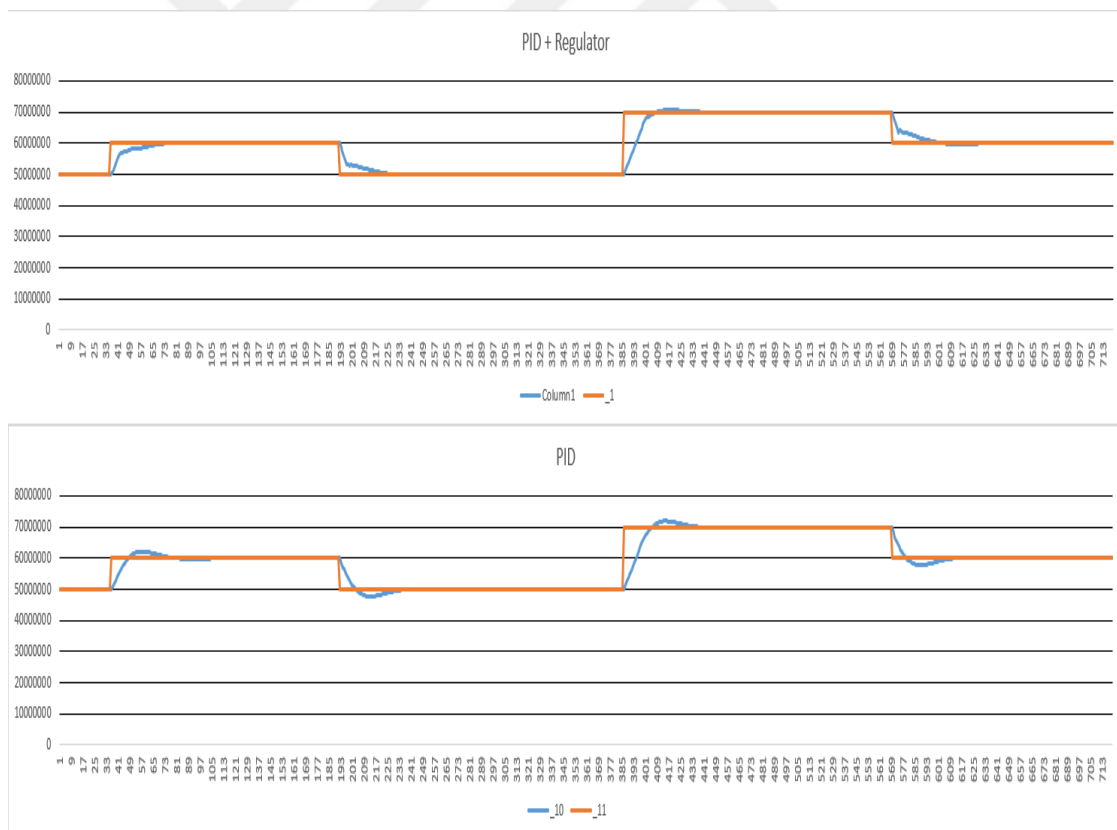
Çıkış “Out” parametreleri, kural tablosunda belirlenen kurallara göre hata ve hata değişim oranlarındaki ayarlanan değerler arasındayken çıkışa direkt etki eden parametrelerdir. Örnek olarak hata değeri E_NL değerinden küçükse ve hata artış değeri ER_NL değerinden küçükse kural tablosunda NL olarak belirtilmiştir. PID kontrolcü çıkışına NL olarak etki edecektir. Buradaki NL değeri Out_NL parametresinden gelir. Out parametreleri; E ve ER parametreleri girildikten sonra çıkışa etki ettiği orana göre artırılıp azaltılarak optimum değerler sağlanmaya çalışılmıştır. Parametre testlerinden sonra ayarlanan değerler Şekil 3.17’ de verilmiştir.

Şekil 3.14: PID_Regülatör Çalışma Parametreleri

■	Out_NL	Real	-10.0	ER_NL	Real	-1.0	E_NL	Real	-3.0
■	Out_NM	Real	-7.5	ER_NM	Real	-0.5	E_NM	Real	-1.5
■	Out_NS	Real	-2.5	ER_NS	Real	-0.1	E_NS	Real	-0.5
■	Out_ZE	Real	0.0	ER_PS	Real	0.1	E_PS	Real	0.5
■	Out_PS	Real	2.5	ER_PM	Real	0.5	E_PM	Real	1.5
■	Out_PM	Real	7.5	ER_PL	Real	1.0	E_PL	Real	3.0
■	Out_PL	Real	10.0						

3.4.4. Simülasyon Sonuçları

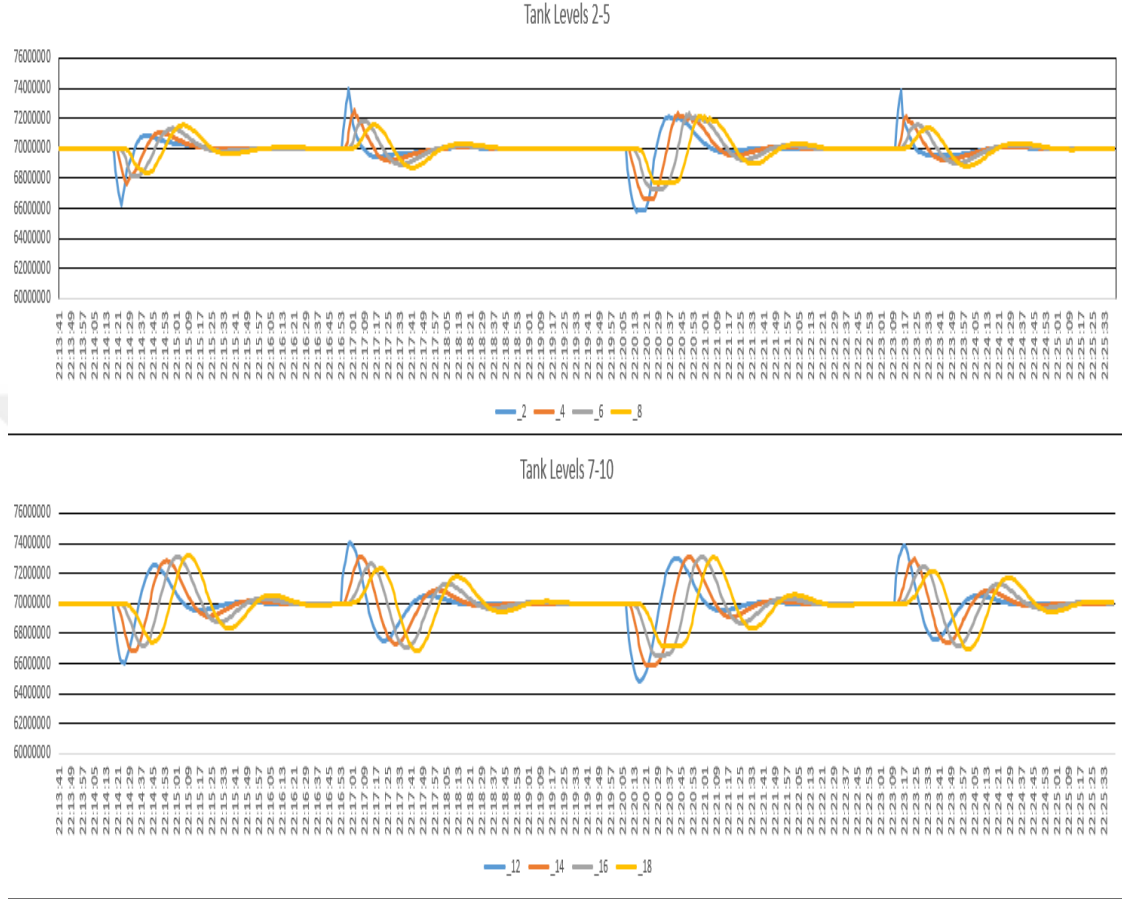
Derlenen sistem, parametreleri girildikten sonra çalıştırılmıştır. Yapılan testte PID Regulator destekli kontrol edilen flotasyon hücrelerinin set değerleri ile Klasik PID ile yapılan Flotasyon hücrelerinin set değerleri aynı zamanda değiştirilerek iki kontrol bloğunun verdiği tepkiler kıyaslanmıştır. Şekil 3.15’de PID Regulator ile Klasik PID kontrolü arasındaki tepki farklılıkları verilmiştir.

Şekil 3.15: PID Regulator ile Klasik PID Arasındaki Fark

Aynı zamanda, birbirine bağlı her flotasyon hücrelerinin kontrolü esnasındaki tepkiler Şekil 3.16’da verilen grafikler ile kıyaslanmıştır. Aşağıdaki grafiklerde ilk örnek

2, 3, 4, ve 5 numaralı hücrelerin seviyesine, altındaki ise 7, 8, 9, 10 numaralı hücrelerin seviyesine aittir.

Şekil 3.16: Flotasyon Hücreleri Arasındaki Kontrol Salınımı



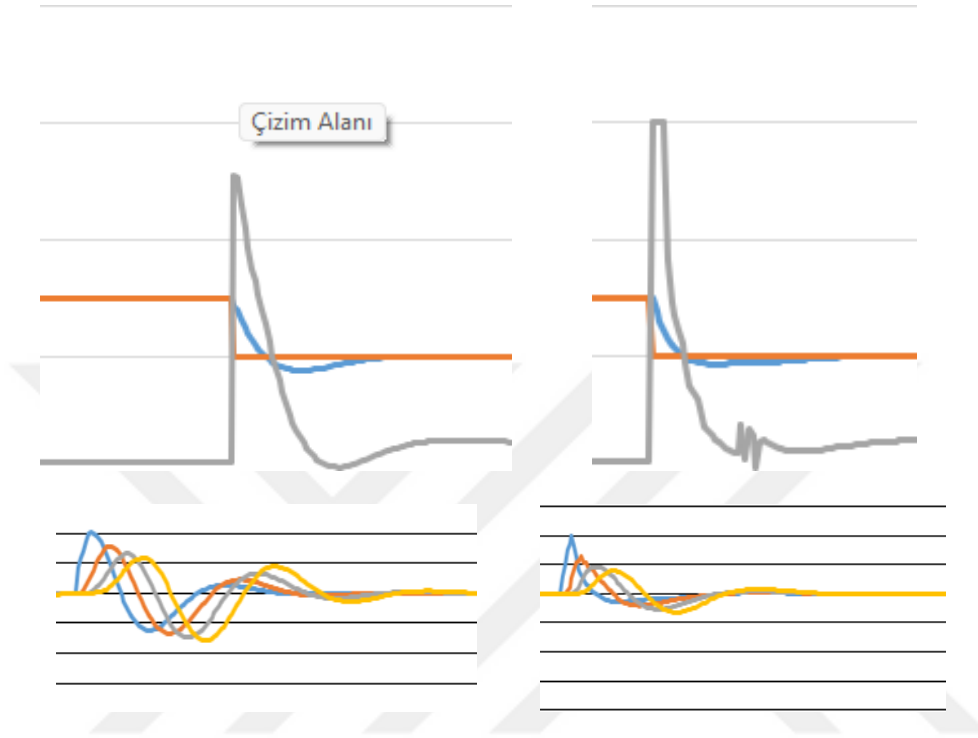
İlk hücredeki salınımların az olması ve ardışıl hücrelerdeki salınımı azaltan faktörler iki sistem arasındaki temel farkları oluşturmuştur. Regüleli işleme tabi tutulmuş flotasyon hücreleri eşit şartlarda diğer hücelere göre daha kısa sürede tepki sağlamıştır, daha az salınım ve dalgalanma yapmıştır.

Geliştirilen PID Regulator bloğu ve bu metotla kontrol edilen sistemin, normal bir PID kontrolörüne göre daha efektif çalıştığı gözlemlenmiştir. Geliştirilen yöntem ile salınımlar düşürülmüş, set değerine ulaşma hızı artmış ve kritik değerlerin dışına çıkması engellenmiştir.

Şekil 3.17’de verilen kontrolcülerin kıyaslamalı grafiğinde, ilk grafik klasik PID kontrolcüsü ile yapılan teste, diğeri ise regüleli işleme girmiş teste aittir.

Turuncu renk proses set değerini, mavi renk flotasyon hücresinin köpük seviyesini, gri renk ise flotasyon hücresi altında bulunan ve köpük seviyesini ayarlayan vananın proses set değerini ifade etmektedir.

Şekil 3.17: Kontrolcülerin Çıkış Enstrümanına Etkisi



Şekil 3.17’de görüldüğü üzere PID kontrolörü salınım yapmak üzereyken regülasyona tabi tutulmuştur ve salınım genliğini azaltmıştır. Çıkış vanasının kontrol değeri PID regülatör sayesinde hızlı bir salınım ile manipüle edilmiştir.

Genel olarak set değeri toleransına regülasyona tabi olan kontrolör daha çabuk ulaşmıştır ve salınımı daha azdır. Ardışıl hücrelerde bu fark daha bariz görülür. Daha olağandışı durumlarda kritik limit aktiviteleri devreye girer ve flotasyon hücresinin PID tepki gecikmesinden kaynaklı aşırı boşalması ve aşırı dolmasının önüne geçer. Bu da sistemin genel toparlanma süresini ciddi anlamda düşürür.



DÖRDÜNCÜ BÖLÜM
ENTEĞRE SİSTEM TASARIMI

4.1. TASARIMIN TEMELİ

Önceki bölümlerde aktarılan bilgiler doğrultusunda temel bir tasarım modeli hazırlanmıştır. Tasarımın temel hedefi mümkün olan en hızlı şekilde, en fazla cevher kazanımının sağlanmasına yönelik bir kontrol sistemi tanımlamaktır. Yanı sıra kontrol sistemlerinin tasarımı ve uygulamasına yönelik pratik ve teorik bilgilerde bu bölüm içerisinde yer alacaktır.

Bölüm 1’de Flotasyon dinamikleri kısmında, köpük bölgesinden savaklara dökülen köpük miktarının aslında köpük hızı olarak nitelendirilebileceğimiz belirtilmişti. Bununla birlikte köpük hızının aslında farklı köpük seviyelerinde ve farklı hava debilerinde aynı şekilde olabileceği fakat bununla beraber köpüğün kalitesinin yani cevher kazanımının da kalitesinin artıp azalabileceği aktarılmıştır.

Flotasyon hücrelerinde köpük seviyesi sabit tutularak, hava debisinin artırılması durumunda köpük kalitesinin düştüğü ve cevher kazanımının azaldığı, diğer bir kontrol değişkeni olan hava debisinin sabit tutularak, köpük seviyesinin artırılması durumunda ise cevher kazanımının arttığı deneysel olarak kanıtlanmıştır.

Bölüm 2’de görüntü işleme teknikleri kullanarak, köpük hızı kantitatif bir değişken olarak ölçülmüştü. Bu değişkenin aslında savaklara aktarılan köpük miktarı ile doğru orantılı olduğu ve dolayısı ile cevher kazanımı ile de doğru orantılı olduğu kabul edilebilir.

Bölüm 3’te endüstriyel kontrolcü olarak nitelendirdiğimiz PLC içerisinde koşturulacak ve standart kontrol metodları na nazaran daha hassas çalışan bir kontrol metodu tasarımı aktarılmıştır. Bu kontrol bloğu flotasyon hücresi hava debisinin ve köpük seviyesinin kontrolü için kullanılacaktır.

Her flotasyon hücresinin kendi içerisinde kapalı çevrim kontrol yaptığı ve ardışık hücreler ile flotasyon bankları kurulmasına karşın bu kontrolün sistem üzerine uygulanabileceği de değerlendirilmiştir.

Nihai olarak tasarımın temeli tek bir flotasyon hücresi özelinde yapılacaktır. Köpük hızı ölçüm sisteminden gelen köpük hızı bilgisi ile operatör tarafından cevherin niteliğine bağlı olarak belirlenecek hız değeri arasında bir model oluşturulup, flotasyon hücresinin hava debisini ve köpük seviyesini optimum şekilde ayarlaması sağlanacaktır. Böylece maksimum düzeyde verim sağlayan kontrol sistemi tasarımı oluşturulacaktır.

Sistem tasarımı yapılır iken kontrol mühendisliğinin temelini oluşturan teorik ve pratik bilgiler ile birlikte kontrol sistemlerinin dokümantasyonuna ait bilgiler de aktarılacaktır.

4.2. KONTROL FİLOZOFİSİ

Otomasyon ve kontrol sistemleri her ne kadar sistemleri otomatik olarak kontrol ederek denetleseler de, her durumda işin ehli operatörlerin dokunuşlarına ihtiyaç duyarlar. Operatörler ilgili ve tecrübe ettiği proses ile alakalı sistem parametrelerini kontrol sistemine bildirir ve prosesi çalıştırır. Kontrol sistemi proses verileri operatörler tarafından izlenir, prosesin verimliliği ya da güvenliği için bir dokunuş gerekiyor ise bunu tekrarlar ve prosesin düzenli şekilde kontrol edilmesi sağlanır. Görsel 4.1’de örnek bir kontrol odası temsili olarak verilmiştir.

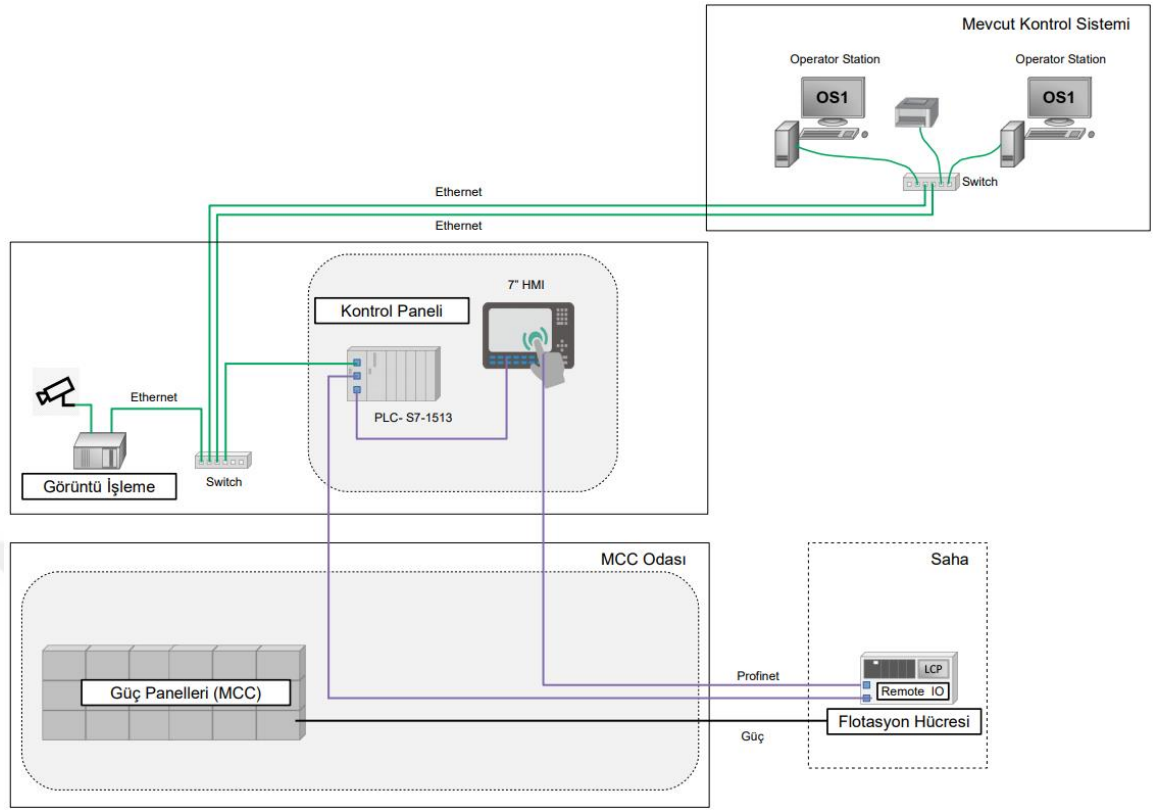
Görsel 4.1: Operatör ve Kontrol Odası



Kaynak: (tiga.us, 2024).

Bu çalışmada kontrol sisteminin parametrelerinin ayarlanabilmesi için HMI kullanılmıştır. Operatör HMI üzerinden veri girişi yaparak, sistemin alarm ve çalışma limitlerini belirleyerek flotasyon hücrelerini idame edecektir. Kontrol sisteminin genel topolojisi aşağıda hazırlanan çizimde Şekil 4.1’de verilmiştir.

Şekil 4.1: Kontrol Sistemi Topolojisi



Kontrol panelinde PLC ve operatörün kontrolleri gerçekleştireceği HMI bulunur, kameralı görüntü işleme sistemi ile bir ethernet switch üzerinden haberleşerek köpük hızının değerini almaktadır. Flotasyon hücresi üzerinde bulunan giriş çıkış bağlantı paneli sahadaki enstrümanlara bağlıdır, aynı zamanda kontrol panelinde bulunan PLC ile haberleşmektedir. Böylece PLC içerisinde koşturulan yazılım sayesinde flotasyon hücresi üzerindeki enstrümanlar senaryo doğrultusunda kontrol edilir.

Tesiste merkezi bir kontrol sistemi olduğu varsayılarak ve tasarlanan sistemin kontrolünün merkezi sistem tarafından da yapılabileceği düşünülerek merkezi DCS (Distributed Control System) te topolojide gösterilmiştir. Aynı zamanda flotasyon hücresi üzerinde bulunan uzak giriş çıkış paneli, HMI ve PLC arasında bir ring sistemi kurularak haberleşme sistemi altyapısının kablo bazında güvenliği sağlanmıştır.

Flotasyon hücresi üzerinde bulunan ve güç ihtiyacı olan ekipmanların MCC (Motor Control Center) üzerinden beslenerek çalıştırılması sağlanmıştır.

4.3. AKIŞ ŞEMASI

Akış şeması algoritmaları ve işlemleri birbirine oklarla bağlı değişik tiplerdeki kutular içerisinde gösteren yaygın bir şema tipidir. Akış şemaları çeşitli alanlardaki işlem ve uygulamaların yönetilmesi, belgelendirilmesi, tasarlanması ve çözümlenmesinde kullanılır (tr.wikipedia.org, 2024).

Tasarım kriterlerinin belirlenmesi sonrasında Şekil 4.2’de verilen, tasarlanacak sistemin akış şeması oluşturulmuştur. Akış şeması genel olarak bir flotasyon hücresinin kontrolünü sağlayacak giriş ve çıkışlara aittir. Tasarımı yapılan flotasyon hücresi kontrol sisteminin en temel hali ile giriş çıkışları aşağıda verilmiştir.

Köpük Seviye Bilgisi:

Hücre üzerindeki seviye sensöründen alınır.

Hava Debisi Bilgisi:

Hücre üzerindeki hava hattı debimetresinden alınır.

Köpük Hızı Bilgisi:

Görüntü İşleme Sisteminden haberleşme protokolü üzerinden alınır.

Köpük Seviye Ayar Çıkışı:

Flotasyon hücresi altında bulunan oransal vana ile sağlanır.

Hava Debisi Ayar Çıkışı:

Hava hattı üzerinde bulunan oransal vana ile sağlanır.

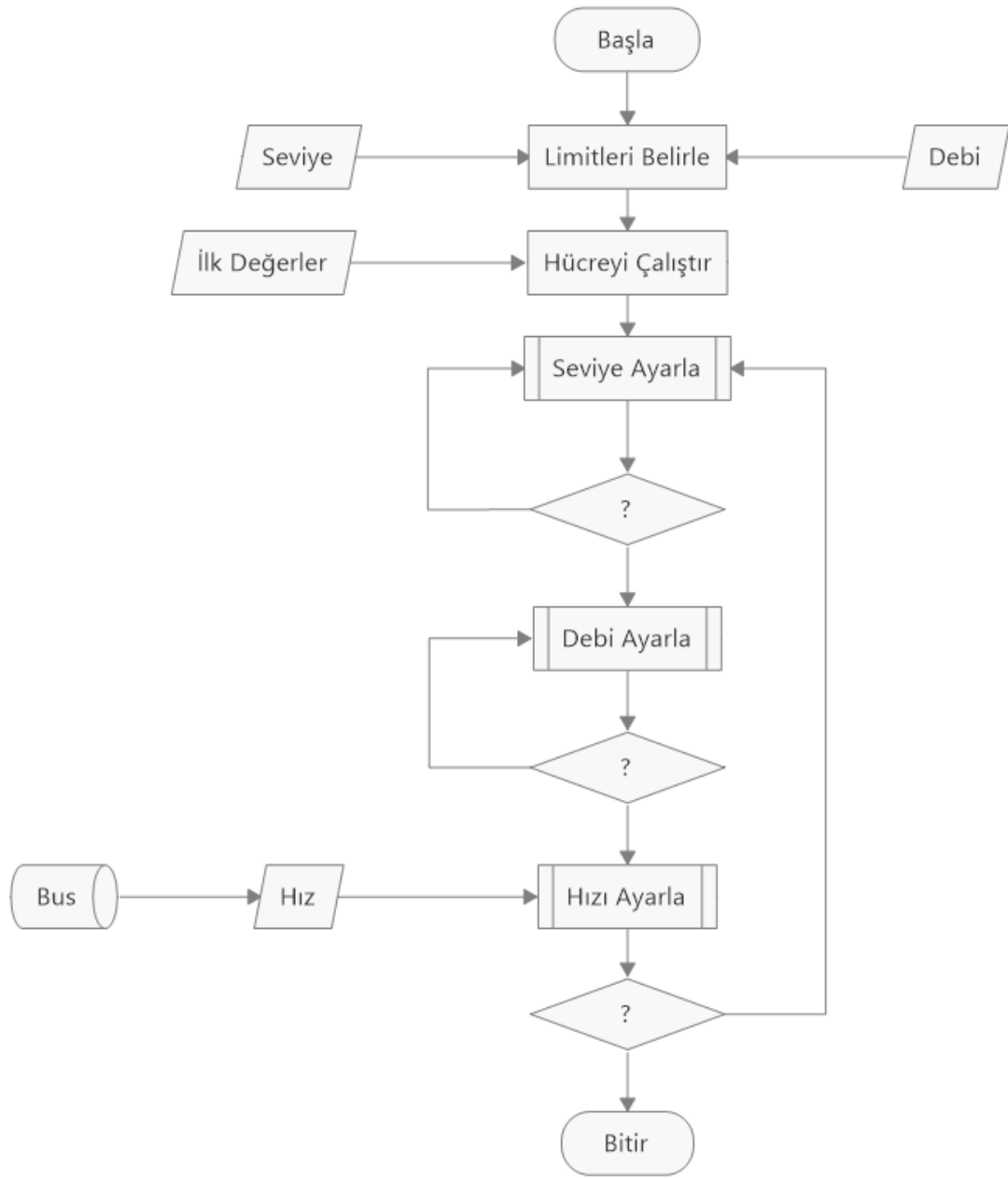
Limitler:

HMI üzerinden operatör belirler.

İlk Değerler:

HMI üzerinden operatör belirler.

Şekil 4.2: Entegre Kontrol Sistemi Akış Şeması



Sistemin nihai olarak, hava debisi ayar vanası ve köpük seviyesini ayarlayan vanası olmak üzere toplam 2 çıkışı bulunmaktadır. Bu vanaların limitler dahilinde yada güvenli bölgelerde çalıştırılması için öncelikli olarak operatör tarafından bu vanaların çalışma limitlerinin belirlenmesi gerekir.

Diğer bir aşamada flotasyon hücresinin doldurulması sonrası ilk çalışma değerleri belirtilmelidir. Bu bilgi operatörler tarafından tecrübesel olarak sisteme tanıtılır. Böylece kontrol vanaları çok fazla salınım yapmadan ilk çalışmalarını yapabilir, bununla beraber flotasyon hücresinin ilk çalıştırılması manuel olarak da yapılabilir.

Öncelikli olarak köpük seviyesi kontrol bloğu ve sonrasında ise hava debisi kontrol bloğu çalıştırılır, bu prosesler alt program olarak, Bölüm 3 te tasarlanan ve detaylı şekilde açıklanan kontrol yazılımını kullanırlar.

Bir sonraki aşamada Bölüm 2 de anlatılan görüntü işleme temelli, harici olarak çalışan ve köpük hızı bilgisini sağlayan sistem tarafından köpük hızı bilgisi alınır. Hız ayarlama kontrol bloğu tarafından ilgili işleme tabi tutularak prosesin sürdürülmesi sağlanır.

4.4. KONTROL SENARYOSU

Flotasyon hücresinin en verimli şekilde çalıştırılabilmesi için, köpük formunun ve akışının düzenli olması yani köpüklerin kaybedilmeden, kazanılmış cevher partikülleri ile birlikte savaklara olan akışın sağlanabilmesi gerekir. Cevher kazanımın en yüksek olduğu durum ise köpük seviyesinin en fazla olduğu durumdur. Bununla birlikte köpük seviyesinin yüksek oluşu köpüklerin daha fazla beklemeleri bu sebeple de kaybedilmeleri anlamına gelir. Burada dengeli bir şekilde köpük transferinin sağlanması gerekir. Kontrol senaryosu akış şemasına uygun olarak şu şekilde çalışacaktır.

Aslında köpük formlarının homojen olarak, kaybedilmeden düzenli bir şekilde savaklara alındığı bir hücre hızı her zaman mümkün görünmektedir, ki bu sebeple flotasyon hücreleri dizaynı yapılmıştır ve endüstride kullanılmaktadırlar. Hedef ise en doğru köpük formunda en uygun hızı yani savaklara olan akışı tespit etmektir. Sistemin çalışması öncesinde tecrübeli bir operatör tarafından sistem için uygun olabilecek bir köpük hızı sisteme girilir. Bununla birlikte seviye ve debi parametrelerinin limitleri de sisteme tanıtılır. Böylece kontrol enstrümanlarının ve kontrol değişkenlerinin alt üst limitleri belirlenir ve sistemin güvenli şekilde çalışması sağlanır.

Flotasyon hücrelerindeki kontrol adım adım yapılmıştır. Bunun sebebi endüstriyel düzeyde büyük bir tank hücreesindeki değişimlerin anlık olarak izlenememesidir. Kontrol sistemi görüntü işleme sistemi tarafından anlık olarak gelen hıza bağlı olarak bir kontrol senaryosu ile çalışır, Operatör tarafından girilen ilk set değerine bağlı olarak en hızlı şekilde sistemin çalıştırılması hedeflenmiştir. Bunun yanı sıra kontrol senaryosu belirli bir süre ile aralıklı olarak çalışmaktadır, sistemin çalışması sonucu verdiği tepkinin, köpük hızına ve kalitesine olan etkisi operatör tarafından değerlendirilmektedir.

Görüntü işleme sisteminden gelen köpük hızı verisi ile sistemin referans aldığı köpük set değeri arasındaki farka göre sistem yükü kontrol edilmiştir. Bu noktada öncelikli olarak köpük seviyesi bir miktar artırılmıştır, bu durumda hava debisi sabit kaldığı için köpük hızında bir miktar düşüş görülmüştür. Sonrasında yine köpük hızı referansına bağlı olarak hava debisi artırılmaya başlanmış ve istenilen hıza gelene kadar aralıklı periyotlar ile bu kontrol tekrarlanmıştır.

Operatör tarafından istenilen köpük formu ve hızı yakalandığında sistemin optimum çalışma aralıkları da o an tespit edilmiştir, bundan sonraki süreçlerde tespit edilen köpük yüksekliği ve hava debisi kullanılarak sistemin en verimli şekilde çalışması sağlanmıştır.

4.5. PLC YAZILIMI

Şekil 4.2’de verilen akış şeması ve kontrol senaryosu doğrultusunda PLC yazılımı geliştirilmiştir. Akış şeması incelendiğinde 3 temel programın oluşturulacağı görülmektedir. Bahsi geçen programlar, “Seviye Ayarla”, “Debi Ayarla” ve “Hız Ayarla” alt programlarıdır.

Bununla birlikte PLC yazılımı içerisinde dijital giriş çıkışların tanımlanması, analog işlemlerin tanımlanması, tag tablolarının oluşturulması, haberleşme parametrelerinin hazırlanması vb. gibi standart kodlamalara da ihtiyaç vardır, fakat bu çalışmanın konusu ileri düzey entegre bir kontrol sistemi tasarımı olması sebebi ile bu konulara detaylı olarak yer verilmemiştir.

Bölüm 3’te, bulanık mantık temelli PID Regülatörü kullanılarak Seviye ve Debi kontrollerinde kapalı çevrim döngüsü ile çalışan alt programın tasarımı ve simülasyonu gerçekleştirilmişti. Entegre kontrol sistemi tasarımının tamamlanması için ise Köpük Hızı ayarının bu kapalı çevrim kontrolcüler ile ilişkilendirilmesi ve sistemin bir bütün olarak çalışması hedeflenmiştir.

Şekil 4.3’de geliştirilen yazılımda tanımlanan değişkenlerin bulunduğu bölümün görseli verilmiştir. Geliştirme öncesi değişkenler tanımlanmış ve sonrasında PLC kodu yazılımı esnasında kullanılmışlardır.

Şekil 4.3: Köpük Hızı Bloğu Yazılım Değişkenleri

	Name	Data type	Default value	Comment
1	Input			
2	frothSpeedSetVal	Real		
3	frothSpeedActVal	Real		
4	frothSpeedToVal	Real		
5	inLevelSetValue	Real		
6	inAirFlowSetValue	Real		
7	callOK	Bool		
8	Output			
9	doCtrl	Bool		
10	outLevelSetValue	Real		
11	outAirFlowSetValue	Real		
12	InOut			
13	<Add new>			
14	Temp			
15	frothSpeedUpperVal	Real		
16	frothSpeedLowerVal	Real		
17	levelCtrlTemp	Bool		
18	mLevelValue	Real		
19	mAirFlowValue	Real		
20	levelInc	Bool		
21	levelDec	Bool		
22	airFlowInc	Bool		
23	airFlowDec	Bool		
24	levelOrAirFlow	Bool		
25	Constant			
26	<Add new>			
27	Return			
28	speed_Ctrl	Void		

Değişkenleri tanımlanması sonrası tolerans değerlerinin ve zamanlayıcıların tanımlandığı bölümün yazılımına geçilmiştir. Şekil 4.4'te yazılımın bu aşaması ile ilgili görsel verilmiştir.

Şekil 4.4: Köpük Hızı Bloğu Yazılım Aşaması 1

```

S7_1500 > PLC_1 [CPU 1513-1 PN] > Program blocks > Ctrls > speed_Ctrl [FC7]
Block interface
IF... CASE... FOR... WHILE... (*...*) REGION
OF... TO DO... DO...

1 // blok dışarıdan aktif edildiğinde işlem yapsın
2 IF #callOK THEN
3 // köpük hızının alt ve üst limitlerinin kontrol için belirlenmesi
4 #frothSpeedUpperVal := #frothSpeedSetVal + #frothSpeedTolVal;
5 #frothspeedLowerVal := #frothSpeedSetVal - #frothSpeedTolVal;
6
7 // 1 dakika boyunca durumların devami varmi ?
8 "upperCheck".TON(IN := (#frothSpeedActVal > #frothSpeedUpperVal),
9 PT := T#60000ms);
10
11 "lowerCheck".TON(IN := (#frothSpeedActVal <= #frothspeedLowerVal),
12 PT := T#60000ms);
13
14 // kontrolcülerin bir kereye mahsus işlem yapması sağlanır
15 "linc_R_TRIG"(CLK:=#levelInc);
16 "lDec_R_TRIG"(CLK:=#levelDec);
17 "afInc_R_TRIG"(CLK:=#airFlowInc);
18 "afDec_R_TRIG"(CLK:=#airFlowDec);
19
20 // giriş değerleri manipule edilecek degere aktarılır
21 #mLevelValue := #inLevelSetValue;
22 #mAirFlowValue := #inAirFlowSetValue;
23
24 // manipule edilen değerler hesaplanır değişiklik %2 seviyesinde
25 IF #doCtrl THEN
26
27 IF "linc_R_TRIG".Q THEN
28 #mLevelValue := #inLevelSetValue + (#inLevelSetValue * 0.02);
29 #outLevelSetValue := #mLevelValue;
30 ELSIF "lDec_R_TRIG".Q THEN
31 #mLevelValue := #inLevelSetValue - (#inLevelSetValue * 0.02);
32 #outLevelSetValue := #mLevelValue;

```

Devam eden aşamalarda köpük hızına bağlı kontrol senaryosuna uygun yazılım gerçekleştirilerek ve Köpük Hızı Ayarlama bloğunun çıkışları tanımlanarak yazılım bitirilmiştir. Şekil 4.5’de yazılımın bu aşaması ile ilgili görsel verilmiştir.

Şekil 4.5: Köpük Hızı Bloğu Yazılım Aşaması 2

```

50      #doCtrl := TRUE;
51      #airFlowDec := TRUE;
52      #levelInc := FALSE;
53      #levelOrAirFlow := FALSE;
54      ELSIF (#frothSpeedActVal > #frothSpeedUpperVal) AND "upperCheck".Q AND NOT #levelOrAirFlow THEN
55      #doCtrl := TRUE;
56      #levelInc := TRUE;
57      #airFlowDec := FALSE;
58      #levelOrAirFlow := TRUE;
59      ELSIF (#frothSpeedActVal <= #frothSpeedLowerVal) AND "lowerCheck".Q AND #levelOrAirFlow THEN
60      #doCtrl := TRUE;
61      #levelDec := FALSE;
62      #airFlowInc := TRUE;
63      #levelOrAirFlow := FALSE;
64      ELSIF (#frothSpeedActVal <= #frothSpeedLowerVal) AND "lowerCheck".Q AND NOT #levelOrAirFlow THEN
65      #doCtrl := TRUE;
66      #levelDec := TRUE;
67      #airFlowInc := FALSE;
68      #levelOrAirFlow := TRUE;
69      ELSE
70      #doCtrl := FALSE;
71      #levelInc := FALSE;
72      #levelDec := FALSE;
73      #airFlowInc := FALSE;
74      #airFlowDec := FALSE;
75      #levelOrAirFlow := TRUE;
76      END_IF;
77
78
79      ELSE
80      // blok girişi aktif edilmez ise girişler manipule edilmeden doğrudan çıkışa yönlendirilir
81      #outLevelSetValue := #inLevelSetValue;

```

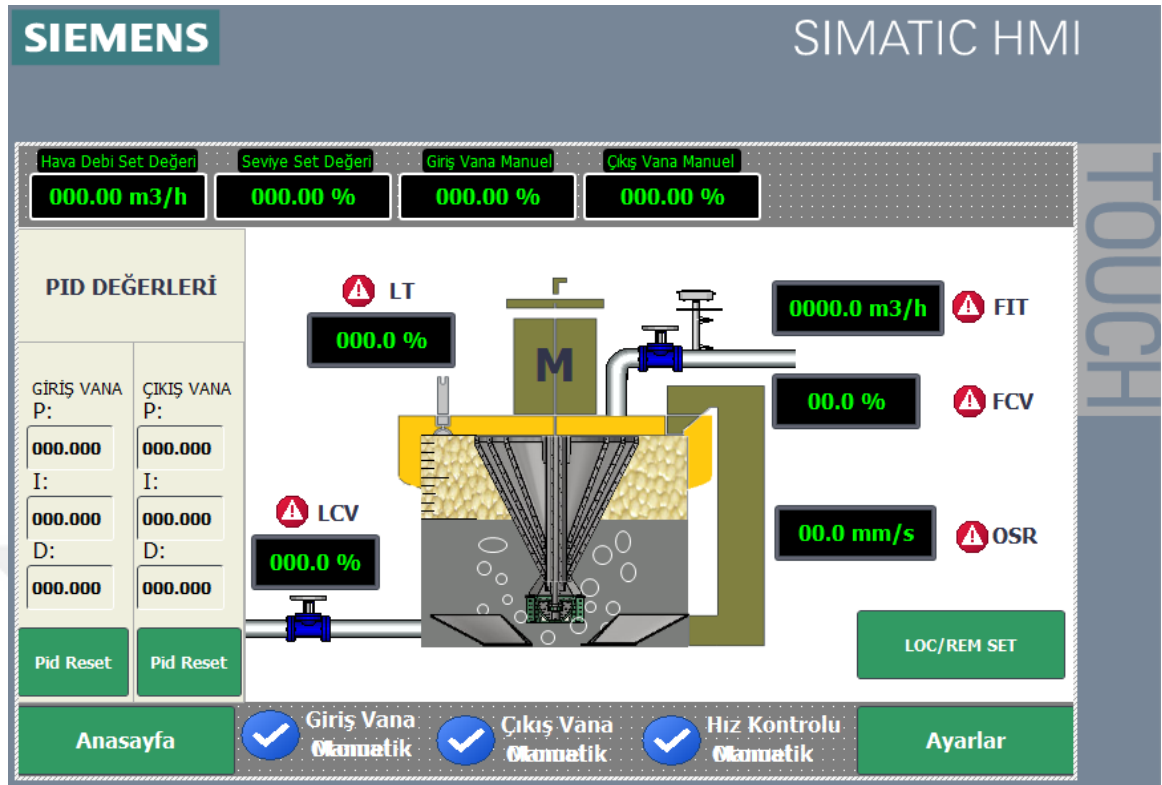
4.6. KONTROL ARAYÜZÜ

Kontrol arayüzü, bir prosten gelen verilerin ekranda grafiksel olarak gösterimini sağlamak ve prosese geri bildirimde bulunan kullanıcı komutlarını sisteme iletmek için endüstriyel proseslerde ve makinelerde kullanılan PC tabanlı veya Operatör Paneller üzerinde çalışabilen insan makine ara yüz sistemidir.

Bu çalışma da, flotasyon hücresinin kontrol arayüzü için, Siemens marka KTP serisi 7" boyutlarında bir operatör panelinin tasarımı yapılmıştır. Operatör paneli PLC ile, Siemens tarafından endüstriye sunulan Profinet haberleşme protokolü üzerinden haberleşmektedir.

Kontrol arayüzü temel olarak 3 sayfa üzerinden tasarlanmıştır. Bu sayfalar Ana Sayfa, Parametre Sayfası ve Alarm Sayfasıdır. Şekil 4.6'da Ana sayfa ekranı verilmiştir.

Şekil 4.6: Kullanıcı Arayüzü Ana Sayfa Tasarımı



Ana Sayfa ekranında orta bölümde flotasyon hücresi grafiksel olarak gösterilmiştir. Proses akışına uygun olarak enstrümanların yerleşimi yapılmıştır. Enstrümanların notasyonları ISA S5 standardına göre yapılmıştır (ANSI/ISA 5.1,2009).

LT (Level Transmitter) : Köpük seviyesi sensörü

LCV (Level Control Valve) : Köpük seviyesini ayarlayan oransal vana

FIT (Flow Indication Transmitter) : Termal hava debimetresi

FCV (Flow Control Valve) : Hava debisini ayarlayan oransal vana

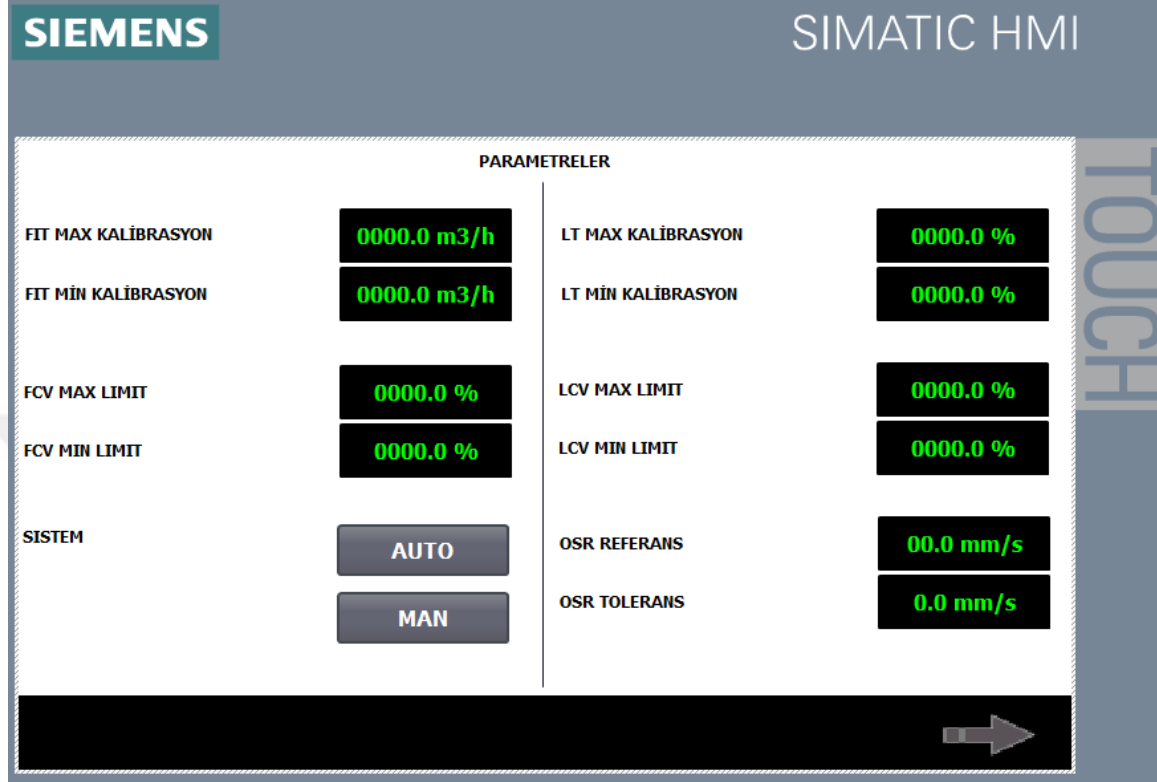
OSR (Optical Speed Recorder) : Köpük hızı ölçüm sistemi

Hücreye üst taraftan bağlanan boru görseli hava hattını temsil eder, bu hat üzerinde termal hava debimetresi ile oransal hava debi kontrol vanasının simgeleri yerleştirilmiştir. Hücrenin alt tarafındaki boru hattı üzerine köpük seviyesini boşaltarak ayarlayan oransal seviye vanası simgesi yerleştirilmiştir. Hücrenin sol üst kısmında ise seviye sensörü görseli bulunmaktadır.

Kullanıcı arayüzü otomatik ve manuel çalıştırmaya uygun, tüm kontrollerin yapılabildiği buton ve indikasyona sahip şekilde tasarlanmıştır. Ayarlar sayfasına geçiş

ile kullanıcı tarafından belirlenmesi gereken parametreler ulaşılır. Şekil 4.7’de Kullanıcı Arayüzü parametre sayfasının tasarımı verilmiştir.

Şekil 4.7: Kullanıcı Arayüzü Ayarlar Sayfası Tasarımı



Kullanıcı bu sayfayı kullanarak enstrümanların çalışma limitlerini belirler, ayrıca sistemin otomatik ya da manuel olarak hangi şekilde kontrol edilmesi isteniyorsa bu bölümde seçilir. Kullanıcı sistemin limitlerini belirledikten sonra prosesi 3 farklı şekilde çalıştırabilir. Bu modlar Manuel, Otomatik ya da Tam Otomatik olarak tasarlanmıştır.

Son kullanıcı arayüzü ise Alarm sayfasıdır. Flotasyon hücresinin çalıştırılması ve proses esnasında karşılaşılabilecek problem ve olumsuzlukların kullanıcıya bildirilmesi için Alarm sayfası tasarımı yapılmıştır. Şekil 4.8’de alarm sayfası tasarımının görseli verilmiştir.

Şekil 4.8: Kullanıcı Arayüzü Alarm Sayfası Tasarımı



Alarm sayfası üzerinde proseste oluşan ve PLC değişkenlerine bağlı olarak kullanıcıya bildirilmesi gereken uyarılar ve alarmlar listelenir. Alarmin hangi tarih ve zamanda ne sebeple gerçekleştiği kayıt altına alınır, operatörün ilgili problemi giderdiği ya da yok saydığı durumlarda gösterilen alarm listeden kaldırılır. Muhtemel bir alarm durumu ile konuya açıklık getirilecek olursa;

Flotasyon hücresinde kullanılan köpük seviye sensörü, 4-20 mA akım çıkışına sahiptir ve bu değeri proses seviyesine bağlı olarak analog olarak gönderen bir transmitter içerir. Seviye sensörü PLC nin Analog giriş modülüne bağlıdır. Sensörde bir problem oluştuğunda örneğin sensör kablosunun kopması durumunda, sinyal seviyesi 4 mA in altına düşecektir. Bu durumda PLC deki yazılım değişkeni bunu anlayarak bir alarm bitini aktif eder, alarm ekranında bu durumla ilgili bir açıklama ve bu durumun gerçekleştiği zaman bilgisi görülecektir.

4.6.1. Çalışma Modları

Flotasyon hücresinin ilk devreye alma aşaması, çalıştırılması, kalibrasyonları, bakıma alınması vb. durumlar göz önüne alındığında, kullanıcının tüm bu durumlarda sistemi rahat şekilde kontrol edebilmesi hedeflenmiştir. Bu sebeple kullanıcının sistemi

çalıştırabileceği, manuel, otomatik ve tam otomatik olmak üzere 3 farklı mod hazırlanmıştır.

Manuel Mod; sistemin ilk devreye alım aşamasında, hücrenin hızlı şekilde doldurulup boşaltılması gereken durumlarda ya da bakım zamanlarında kullanılır. Bu mod da oransal vanalar operatörün belirttiği değerlerde açılır ya da kapatılırlar. Kapalı çevrim otomatik bir kontrol döngüsü uygulanmaz.

Otomatik Mod; sistemin köpük hızından bağımsız, operatörün tecrübesel verileri ile belirlediği köpük seviyesi ve hava debisi değeri ile çalışır. Bu modda giriş enstrümanlarından alınan veriler ile oransal vanalar kapalı çevrim kontrol yaparlar.

Tam Otomatik Mod; otomatik moda ek olarak, otomatik modda belirlenen köpük seviyesi ve hava debisi değeri, köpük hızına bağlı olarak manipule edilir ve en hassas değerlerin bulunması sağlanır. Sistemin uzak istasyondan çalıştırılması durumunda kamera üzerinden köpük formu kontrol edilebilir ve operatörün flotasyon hücresi yakınına gelmeden sistemin kontrol edilmesi gerçekleştirilir.

SONUÇ VE DEĞERLENDİRME

Flotasyon prosesinin en temel amacı şüphesiz ki köpük oluşumunu sağlayabilmektir. Tesislerde istenilen köpük formunun sağlanması operatörlerin bilgi, tecrübe ve gözlemlerine bağlı olarak enstrümanlar aracılığıyla yapılır.

Flotasyon tesislerinde işleyiş HMI veya SCADA arayüzü ve PLC tabanlı kontrol sistemleri ile yapılmaktadır. Operatörler deneyimleri ve bilgileri doğrultusunda proses parametrelerini değiştirerek tesislerin işleyişini sağlarlar. Tasarım aşamasında doğru enstrüman seçimi ve montajı yapıldıktan sonra çalışmayı etkileyecek faktör, operatörün uygun hava debisi sağlayarak köpük ve köpük seviyesini ayarlamasıdır. Köpük bölgesinde oluşan köpüğün seviyesi, hacimsel olarak büyüklüğü/küçüklüğü, köpüğün yönü ve hızı gibi durumlar verimi etkilemektedir. Oluşturulan köpüğün olması gereken formda olmaması cevhere erişimi engellemekte, buna bağlı olarak da hedeflenen verime ulaşılmasına etki etmektedir.

Enstrümanların ve ekipmanların seçimi, montajlarının doğru yapılması, kullanılacak kimyasal maddelerin doğru seçilmesi ve oluşabilecek arızalar dışında, işleyişe etki eden tek faktör operatördür. Günümüzde insan faktörünü işleyişten çıkarmak, işleyişteki aksaklıkları ortadan kaldırmakta, hata oranlarını azaltmakta ve verimi arttırmaktadır. Bu çalışmada da flotasyon işleminde operatör faktörü etkilerinin minimize edilmesi amaçlanarak, hataların azaltılması ve verimin artırılması amacıyla tasarım yapılmıştır.

Görüntü işleminin oldukça yaygınlaşması ve birçok farklı sektörlerde kullanılıyor olması bu çalışmada kullanılması adına ön plana çıkmıştır. Görüntü işleme algoritmalarının çalıştırılarak denenebilmesi için, çalışan bir flotasyon hücresinden alınan video görüntüleri kullanılmıştır. Alınan görüntüler, köpük formunun ve akışının bozulmadan yani köpüklerin kaybedilmeden ve gerçek zamanlı olarak çalışmasını simüle ederek işlenmiştir.

Görüntü işleme sonrasında, bulanık mantık kontrolörü tasarlanarak, gerçek işleyiş parametrelerine bağlı kurallar oluşturulmuş ve bu kurallar çerçevesinde PLC kontrol sistemine aktarılmıştır. PLC içerisine aktarılan veriler mevcut PID bloğunun çıkışına etki ederek ve kontrol sistemi manipule edilerek güçlendirilmiştir. Köpük seviyesinin ve hava debisinin kontrollü bir şekilde artırılıp azaltılması sağlanmış olup stabil, homojen dağılımda ve homojen boyutta köpükler elde edebilecek kontrol

parametrelerinin oluřtuđu gözlenmiřtir. Bu çalıřma neticesinde, verilere bađlı olarak sistemin sürekliliđinin sađlanabileceđi ve operatörün gözlemi/bilgisine dayalı olarak girilen parametrelerle aynı verilerin alındıđı gözlemlenmiřtir.

Çalıřmalar sonucunda, flotasyon iřleminde görüntü iřleme ve bulanık mantık yöntemiyle operatöre bađlı olmadan, gerçek zamanlı olarak sistemin izlenmesi ile köpük seviyesi ve kalitesinin ayarlanabileceđi, oluřan deđiřimlere daha hızlı müdahale edilerek cevher kayıplarının önüne geçilebileceđi ve verimin artacađı öngörölmüřtür.

Ayrıca bu çalıřma, mevcut otomasyon sistemi bulunan tesislerdeki PLC kontrol sistemine entegre olabilmesi ile birlikte, tesis için herhangi bir revizyon veya mevcut yapının bozulması gibi durumları ortaya çıkarmayacak, iřleyen sistemde herhangi bir deđiřiklik olması gerekmeyecektir. Operatör kontrolünün yerini, görüntü iřleme sisteminin alıyor olması, iřletme adına maliyet olarak da avantaj sađlayacaktır.

Bu çalıřmanın cevher çıkarma iřlemine katkı sađlamasının yanı sıra iřletmeler tarafından da talep edileceđi veya kullanılacađı öngörölmektedir.

KAYNAKÇA

- Akış Şeması. (2024, 10 Nisan). Erişim Adresi: https://tr.wikipedia.org/wiki/Akış_şeması
- ANSI/ISA 5.1. (2009, 10 July). *Instrumentation symbols and identification*. Retrieved from: https://webstore.ansi.org/standards/isa/ansiisa2009?gad_source=1&gclid=EAIaIQobChMI5JSKwqK9hwMVdKdoCR0Hag66EAAYASAAEgK6VPD_BwE.
- Babu, R., & Swarnalath, R. (2017). Comparison of different tuning methods for pH neutralization in textile industry. *Journal of Applied Sciences*, 17(3), 142–147.
- Barron, J. L., & Fleet, D. J., & Beauchemin, S. S. (1994). Performance of optical flow techniques. *Int. Journal of Computer Vision*, 12, 43-77.
- Barron, J.L., & Thacker, N.A. (2004, 15 May). *Tutorial: Computing 2D and 3D optical flow*. Retrieved from: https://faculty.runi.ac.il/toky/old_courses/compphoto-09/handouts/2004-012.pdf.
- Berthold K.P. Horn, & Brian G. Schunck. (1980). *Determining optical flow* (1st ed.). London: Artificial Intelligence Laboratory.
- Bhondayi, C. (2020). Flotation froth phase bubble size measurement. *Mineral Processing and Extractive Metallurgy Review*, 43(2), 251-273.
- Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8), 790-799.
- Comaniciu, D., & Meer, P. (1997). Robust analysis of feature spaces: Color image segmentation. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 17(19), 750-755.
- Dilek, B.G. (2012). *Video görüntüleri içinden hareketli nesne ayıklanması ve izlenmesi* (Yüksek Lisans Tezi). Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.
- Flotasyon Bankı. (2024, 7 Mart). Erişim Adresi: <https://www.ypt.com.tr>.
- Flotasyon Hücresi. (2024, 3 Ocak). Erişim Adresi: <https://www.ypt.com.tr>.
- Fukunaga, K., & Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1), 32-40.

- IEC 61131-3. Programmable Controllers – Part 3: Programming Languages. (2013, 4 April). Retrieved from: <https://intweb.tse.org.tr/Standard/Standard/Standard.aspx?>
- Lucas, B.D., & Kanade, T. (1981,15 April). *An iterative image registration technique with an application to stereo vision*. Retrieved from: https://www.researchgate.net/publication/215458777_An_Iterative_Image_Registration_Technique_with_an_Application_to_Stereo_Vision_IJCAI.
- Oparetör Ekranı. (2024, 8 Mart). Erişim Adresi: <https://www.tiga.us/blog/what-is-scada-and-how-it-increases-efficiency>.
- Ostadrımı, M., & Farrokhpay, S., & Gharibi, K., & Dehghani, A. (2019). Estimating bubble loading in industrial flotation cells. *Minerals*, 9(4), 222-232.
- Ostadrımı, M., & Farrokhpay, S., & Pirmoradi, S., & Noparast, M. (2022). Effect of air flow rate and froth depth on the flotation performance: An industrial case study in a 10 m³ cell. *Physicochem. Probl. Miner. Process*, 58(5), 1-8.
- Özüntürk, E. (2007). *Optik akış ile hareket tespiti* (Yüksek Lisans Tezi). Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Zafir, E. ve Ayşe, N. Ş. (2006). Köpük görüntüsü ve flotasyon performansı arasındaki ilişkinin görüntü analiz sistemi ile incelenmesi. *Madencilik Dergisi*, 45(2), 27-38.
- Zeynep, Y. H. (2023). Kontrol parametrelerinin sürekli frekans salınım yöntemleri ile ayarlanmasının nötralizasyon prosesi örneğinde geri besleme kontrol performansına etkisinin incelenmesi. *ESOGÜ Müh. Mim. Fak. Dergisi*, 31(3), 795-800.

DİZİN

-A-

Algoritma, 41

Analiz, 1, 15, 37, 41

ANSI, xv, 66, 72

Aperture, 18

-B-

Bulanık Mantık, v, 39, 43

-C-

Cevher, 1, 61

Computer Vision, vi, xv, 72

Control System, 58

CPU, 12, 35, 36

CV, xv, 27

-E-

Endüstriyel Otomasyon, v

EthernetIP, 12

-F-

Factory IO, 37, 38, 46, 47, 48

Flotasyon, iii, iv, v, 1, 3, 4, 5, 6, 7, 8, 9, 11, 12, 15, 16, 25, 26, 35, 39, 47, 51, 52, 53, 56, 58, 59, 61, 67, 68, 73

Flotasyon Bankı, 4

Flotation, vi

FTE, xv, 12

Fuzzy Logic, vi

-G-

Görüntü İşleme, iii, iv, v, 59

-H-

Haberleşme, 12

Hava debisi, 4, 5, 6, 7, 8, 12, 35, 60, 61, 62, 69

HMI, xv, 12, 57, 58, 59

-I-

Industrial Automation, vi

ISA, xv, 66

-İ-

interlock, 12

-K-

Kontrol Filozofisi, 57

Kontrol Odası, 57

Kontrol paneli, 9

Kontrol Sistemi, iii, iv, 58, 60

Kontrolör, 39

Köpük görüntüsü, v, 1

Köpük hızı, v, 4, 15, 28, 32, 56, 61, 62

-L-

Lucas-Kanade, v, vi, 1, 15, 21, 22, 23, 24, 28

-M-

Madencilik, v

Mil, 3

ModbusTCP, 12, 33

Modernizasyon, 12

-O-

Oksitem, v, vii

OPC Server, 37, 38

OpenCV, 27

-P-

PID, v, vi, xv, 1, 35, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 62

PLC, v, vi, xv, 9, 11, 12, 13, 32, 35, 36, 37, 38, 41, 44, 46, 47, 56, 58, 62, 65, 68

ProfiNET, 12

Proses, v, 1, 3, 4, 12, 15, 26, 35, 39, 40, 41, 42, 44, 47, 48, 54, 57, 67, 68

Python, 27

-R-

Regulator, 41, 43, 44, 45, 46, 52, 53

Ring, 12, 58

-S-

SCADA, xv, 12

Sensör, 10

Seviye kontrolü, 3, 7

Shi-Tomasi, 24, 28

Siemens, v, vi, 35, 36, 38, 65

Star, 12

-T-

TIA, 36, 38, 39, 41, 42

Topoloji, 12

-V-

Video, 16, 25, 26, 27, 28