



**ROBOTİK MONTAJ SİSTEMLERİNİN
PETRİ AĞI
TEMELLİ KONTROLÜ**

YASEMİN İÇMEZ

**YÜKSEK LİSANS TEZİ
MEKATRONİK MÜHENDİSLİĞİ ANA BİLİM DALI**

YRD. DOÇ. DR. GÖKHAN GELEN

**2015
Her hakkı saklıdır.**

**T.C.
GAZİOSMANPAŞA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANA BİLİM DALI**

YÜKSEK LİSANS TEZİ

**ROBOTİK MONTAJ SİSTEMLERİNİN
PETRİ AĞI
TEMELLİ KONTROLÜ**

YASEMİN İÇMEZ

**TOKAT
2015**

Her hakkı saklıdır

Yasemin İÇMEZ tarafından hazırlanan “**Robotik Montaj Sistemlerinin Petri Ağı Temelli Kontrolü**” adlı tez çalışmasının savunma sınavı 17.06.2015 tarihinde yapılmış olup aşağıda verilen Jüri tarafından Oy Birliği İle Gaziosmanpaşa Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Ana Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Danışman

Yrd. Doç. Dr. Gökhan GELEN
Gaziosmanpaşa Üniversitesi

Üye

Doç. Dr. Ulaş EMİNOĞLU
Niğde Üniversitesi

Üye

Doç. Dr. Mehmet AKAR
Gaziosmanpaşa Üniversitesi



Gaziosmanpaşa Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
tarih ve sayılı kararıyla onaylanmıştır.

ONAY



Prof. Dr. Mehmet Ali SAKİN

Fen Bilimleri Enstitüsü Müdürü

10/07/2015

TEZ BEYANI

Tez yazım kurallarına uygun olarak hazırlanan bu tezin yazılmasında bilimsel ahlak kurallarına uyulduğunu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezin içerdığı yenilik ve sonuçların başka bir yerden alınmadığını, kullanılan verilerde herhangi bir tahrifat yapılmadığını, tezin herhangi bir kısmının bu üniversite veya başka bir üniversitedeki başka bir tez çalışması olarak sunulmadığını beyan ederim.

YASEMİN İÇMEZ

HAZİRAN 2015

ÖZET

YÜKSEK LİSANS TEZİ

ROBOTİK MONTAJ SİSTEMLERİNİN PETRİ AĞI TEMELLİ KONTROLÜ

YASEMİN İÇMEZ

**GAZİOSMANPAŞA ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANA BİLİM DALI**

TEZ DANIŞMANI: YRD. DOÇ. DR. GÖKHAN GELEN

Robotik montaj sistemleri günümüz endüstrisinde yaygın biçimde kullanılmaktadır. Montaj planlaması ve robotik görev planlaması içeren robotik montaj kontrolörlerinin sentezi ve gerçekleştirilmesi oldukça zor bir konudur. Bu tipteki sistemler Ayrık Olay sistemleri olarak sınıflandırılmaktadır. Bu tip sistemler için gözeticilerin hesaplanması ve gerçekleştirilmesi sıklıkla rastlanılan araştırma konularındandır. Robotlar tarafından gerçekleştirilen görevlerin artmasıyla, robotun yerine getireceği görevlerin kontrolü, gerçekleştirilmesi ve özelleştirilmesinde formal yöntemlere olan ihtiyaç artmaktadır. Yüksek Lisans Tezinin amacı endüstriyel robotlar için formal bir yöntem olan Petri ağlarına dayalı kontrol kodu elde etmektir. Bu tezde montaj birimi üreten Mitsubishi RV-2SDB Robotu üzerinde deneysel çalışmalar yapılmaktadır. Bu birim, bir adet taban, piston, yay ve kapağın montajlanmasından oluşmaktadır. İlk olarak Petri ağları kullanılarak montajlanacak hücre modeli oluşturulur. Sonrasında sistemin kontrol spesifikasyonları oluşturularak mevki değişmezi tabanlı gözetici hesabı yapılmaktadır. Bu tez Petri ağı tabanlı gözetici robotik montaj birimi kontrolünü gerçek zamanlı olarak gerçekleştirebilmek için tasarlanmış ve uygulanmıştır. Bu çalışma bir robot kontrolöründe Petri ağı gözeticisinin uygulanmasını ve robot programlama da formal bir yöntem olarak kullanılmasını sağlamaktadır.

2015, 68 SAYFA

Anahtar Kelimeler: Petri ağları, Gözetimli kontrol teorisi, Montaj robotu, Melfa Basic programlama yazılım dili.

ABSTRACT

MASTER THESIS

PETRI NET BASED CONTROL OF ROBOTIC ASSEMBLY SYSTEMS

YASEMİN İÇMEZ

GAZIOSMANPASA UNIVERSITY INSTITUTE OF SCIENCE

DEPARTMENT OF MECHATRONICS ENGINEERING

SUPERVISOR: ASST. PROF. DR. GÖKHAN GELEN

Robotic assembly systems are widely used in nowadays industry. The synthesis and implementation of robotic assembly controllers is a difficult issue, which involves assembly planning and task planning. These types of systems are classified as Discrete Event Systems (DES). The computation and implementation of supervisors for such systems are popular research topics. As the number of tasks accomplished by robots is highly increasing, it is required to have formal methods of specifying, controlling and implementation of the tasks that a robot should be able to perform. The aim of this M. Sc. Thesis is to develop a general methodology to obtain control code for industrial robots based on Petri nets as a formal tool. In this thesis, an assembly cell consists of a Mitsubishi RV-2SDB Robot manipulator is considered to perform experimental studies. This cell produces an assembled component that consists in a base, a pine, a spring and a cover. Initially, the assembly cell is modeled by using Petri net formalism. After constructing of control specifications, the supervisor can be synthesized by using a well-known place invariant-based method. This PN based supervisory structure is designed and implemented to carry out a real-time control of robotic assembly cell. In the sense of practice, the implementation of a Petri net supervisor in a robot controller provides to use of formal methods in robot programming.

2015, 68 pages

Key Words: Petri nets, Supervisory control theory (SCT), Assembly robotic systems, Melfa basic programming language.

ÖNSÖZ

Yüksek lisans çalışmam boyunca, her türlü konuda yardımlarını esirgemeyen, değerli fikirleriyle çalışmama yön veren kıymetli hocam Sayın Yrd. Doç. Dr. Gökhan GELEN' e, eşim Öğr. Gör. Bilal Okan İÇMEZ' e ve hayatım boyunca desteklerini esirgemeyen aileme ve arkadaşlarıma teşekkürlerimi sunarım.

YASEMİN İÇMEZ

HAZİRAN 2015

İÇİNDEKİLER

Sayfa

TEZ BEYANI-----	ii
ÖZET-----	iii
ABSTRACT-----	iv
ÖNSÖZ-----	v
İÇİNDEKİLER -----	vi
KISALTMALAR -----	viii
ŞEKİLLER LİSTESİ-----	xx
ÇİZELGELER LİSTESİ-----	x
1. GİRİŞ-----	1
2. KURAMSAL TEMELLER -----	5
2.1. Petri Ağları -----	5
2.1.1. Basit Petri Ağları -----	5
2.1.2. Basit Bir Petri Ağının Tetiklenmesi -----	8
2.1.3. Petri Ağlarının Özellikleri -----	8
2.1.4. Petri Ağların Cebirsel Özellikleri -----	9
2.1.4.1 Petri Ağlarında Çakışıklık (Incidence) Matrisi-----	9
2.1.4.2. Petri Ağlarında İşaretleme Vektörleri -----	11
2.1.4.3. Petri Ağlarında Mevki ve Geçiş Değişmezleri -----	11
2.2. Gözetimli Kontrol (Supervisory Control) -----	12
3. MATERYAL VE YÖNTEM -----	14
3.1. Mevki Değişmezleri Kullanılarak Gözetici Sentezlenmesi -----	14

3.2. Robot Montaj İstasyonu -----	18
3.3. Mitsubishi RV- 2 SDB Robotunun Genel Özellikleri -----	20
3.4. Melfa Basic Programlama Dili -----	21
3.4.1. Robot Hareketleri -----	22
3.4.1.1. Lineer Hareket-----	22
3.4.1.2. Noktadan Noktaya Hareket-----	22
3.4.2. Hareket Komutları -----	23
3.4.3. Tanımlama Komutları -----	23
3.4.4. Haberleşme Komutları -----	24
3.4.5. Çoklu iş (Multitasking) Komutları -----	24
3.4.6. Programlama Komutları -----	25
4. BULGULAR VE TARTIŞMA -----	26
4.1. Sistem Petri Ağı Modeli -----	26
4.2. Sistem Gözeticisinin Hesaplanması-----	31
4.3. Petri Ağı Modelinin Melfa Basic Programlama Dili ile Gerçekleştirilmesi -----	46
4.4. Robotik Montaj İstasyonu PA Modelinin Gerçekleştirilmesi -----	49
5. SONUÇ -----	62
KAYNAKLAR -----	63
ÖZGEÇMİŞ-----	66

KISALTMALAR DİZİNİ

Kısaltmalar	Açıklama
AOS	Ayrık Olay Sistemleri
SDM	Sonlu Durum Makinaları
PA	Petri Ağları
Sp	Spesifikasyon
SCT	Gözetimli Kontrol Teorisi

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2. 1. Basit bir Petri ağı modeli -----	7
Şekil 2. 2. Basit bir Petri ağında (a) ilk işaretleme (b) t1 tetiklemesinden sonraki işaretleme -	8
Şekil 2. 3. Örnek Petri ağı modeli -----	10
Şekil 2. 4. Kontrol aşamaları -----	12
Şekil 3. 1. Mevki değişmezleri kullanılarak gözeticisi hesaplanacak örnek petri ağı modeli-	16
Şekil 3. 2. Mevki değişmezleri kullanılarak hesaplanan petri ağı modeli ve C kontrolörü ----	18
Şekil 3. 3. Deney düzeneğinin resmi-----	19
Şekil 3. 4. Deney düzeneği ile montajı yapılan parçalar -----	20
Şekil 3. 5. RV –2 SDB Robotu -----	21
Şekil 4. 1. Tez kapsamında gerçekleştirilecek çalışmalar -----	27
Şekil 4. 2. Robot veri istasyonu genel şeması -----	28
Şekil 4. 3. Robot montaj istasyonunun Petri ağı modeli -----	30
Şekil 4. 4. C1 kontrol mevkisi ve ilgili geçişleri -----	35
Şekil 4. 5. C2 kontrol mevkisi ve ilgili geçişleri -----	37
Şekil 4. 6. C3 kontrol mevkisi ve ilgili geçişleri -----	39
Şekil 4. 7. C4 kontrol mevkisi ve ilgili geçişleri -----	41
Şekil 4. 8. C5 kontrol mevkisi ve ilgili geçişleri -----	43
Şekil 4. 9. C6 kontrol mevkisi ve ilgili geçişleri -----	45
Şekil 4. 10. Robot montaj istasyonunun kontrol edilmiş PA modeli-----	46
Şekil 4. 11. Örnek PA modeli-----	48
Şekil 4. 12. PA modelinin melfa basic ile gerçekleştirilirken kullanılacak program yapısı ----	49

ÇİZELGELER LİSTESİ

Sayfa

Çizelge 4.1. Robotik görevler ve açıklamaları -----	31
Çizelge 4.2. Petri ağı modelindeki mevkilerin açıklamaları-----	31
Çizelge 4.3. Petri ağı modelindeki geçişlerin açıklamaları -----	32

1. GİRİŞ

Bilim ve teknoloji alanında son yıllarda hızlı gelişmeler bilgisayar teknolojisinin gelişimini ivmelendirmiştir. Elektronik ve bilgisayar bilimlerindeki bu hızlı gelişmeler modern endüstrinin ihtiyacında da kullanılmaya başlanarak robotik ve endüstriyel otomasyon sistemlerinin ortaya çıkmasını sağlamıştır.

Robotik ve endüstriyel otomasyon sistemlerinin temelini oluşturan unsur ise *Robotlar*, yani otonom veya önceden programlanmış görevleri yerine getirebilen elektro-mekanik cihazlardır [Anonim, 2015]. Robot kelimesi ilk olarak insan benzeri makineleri akla getirirse de aslında doğrudan bir operatör ya da bağımsız olarak bir bilgisayar programının kontrolünde çalışan bir manipülator olarak günümüz endüstrisinde her alanda karşımıza çıkar. Özellikle otomotiv endüstrisinde çok sayıda robot kullanılır. Robotların endüstriyel kullanım alanları şu şekilde sıralanmaktadır [Arıkan, 2013]; robotların yükleme ve boşaltılması ve malzeme manipülasyonu ve istifleme. Robotlar, bu kullanımların yanı sıra şu uygulamalar için de kullanılmaktadır; deri ile ilgili işlemler, ayakkabı imalatı, kauçuk ile ilgili işlemler, asbest ile ilgili işlemler, gıda maddeleri ile ilgili işlemler, kil ve çimento kullanılarak yapılan imalatlar, cam endüstrisi, giyim endüstrisi, ağaç endüstrisi vb.

Endüstride üretim yapan tesislerde robotlar bir program çerçevesinde çalışır ve insanlar tarafından yapılan ya da kontrol edilen birçok işi bu program dâhilinde yaparlar. Bu birlikte çalışma *Otomasyon Sistemi* olarak ifade edilmektedir. Robotik ve endüstriyel otomasyon sistemlerinin davranışı çoğunlukla ayrık (discrete) olaylarla belirlenir. Bu şekilde olan sistemler ayrık olay sistemleri (AOS) (Discrete Event Systems-DES) veya ayrık olay dinamik sistemleri (Discrete Event Systems-DEDS) olarak adlandırılmaktadır [Cassandras ve Lafortune, 1999].

AOS'ler günümüz modern endüstriyel sistemlerinin modellemesinde, analizinde, tasarımında ve kontrolünde kullanılan yeni bir araştırma konusu olarak ortaya çıkmıştır. Bu yaklaşım öncesinde karşılaşılan basit problemlerin çözümünde ilk olarak sezgisel metotlar kullanılmıştır. Hatta bu eğilim halen güncelliğini korumaktadır. AOS'lerin Kontrol ve otomasyon problemlerinin, kontrol mühendislerinin deneyime dayalı yöntemleriyle çözülmesine çalışılmaktadır. Ancak AOS'lerin gün geçtikçe daha da karmaşık hale gelmesiyle bu tip sistemlerin denetiminde kullanılmak üzere etkin formal yöntemlere olan ihtiyaç hızla artmaktadır.

Bu ihtiyacın karşılanması amacıyla, AOS'lerle ilgili yapılan çalışmalar genel olarak dört farklı yöntemle gerçekleştirilmektedir. Bu yöntemler; Otomata, Petri Ağları (petri nets), Minimaks, Sıralama Ağları ve diğer Cebirsel yöntemlerdir. Literatürde AOS'ler ile ilgili yapılan çalışmalar dikkate alındığında otomata ve petri ağları kullanılarak yapılan çalışmaların diğer yöntemler kullanılarak yapılan çalışmalara göre daha ön planda olduğunu görmek mümkündür.

AOS'lerin kontrolünde yaygın olarak Petri ağları kullanılmaktadır. Grafiksel ve matematiksel bir araç olarak Petri ağları, AOS'lerin modellenmesinde, analizinde, tasarımında ve kontrolünde artan bir hızda kullanılmaya devam etmektedir [Zhou ve DiCesare, 1993]. Petri ağları, otomata ile haberleşme çalışmaları için ağ şeklinde matematiksel bir araç ortaya koyan Alman matematikçi Carl Adam Petri' den ismini almıştır [Petri, 1963]. O zamandan bu yana bilgisayar bilimi, üretim sistemleri, haberleşme sistemleri, endüstriyel otomasyon sistemleri vb. gibi çeşitli alanlarda araştırmalar gerçekleştirilmiştir [Zurawski ve Zhou, 1994].

Petri ağları, eş zamanlılık, senkronizasyon ve kaynak paylaşımını içeren durumların modellenmesini ve oluşabilecek durumların göz önünde canlandırılmasını mümkün kılmaktadır. Bu AOS'lerin modellenmesinde, analizinde, simülasyonunda ve kontrolünde petri ağlarının çok büyük kolaylıklar sağlayacağını aynı zamanda bir ispatıdır [David ve Alla, 1994 - Desrochers ve AlJaar, 1995]. Ancak sıradan Petri ağları, karmaşık sistemleri ifade etmekte her zaman yeterli olamaz. Örneğin, birçok benzer etkileşimli aktivitelerden oluşan karmaşık endüstriyel sistemler modellenirken sıradan Petri Ağları modelinin grafiksel karmaşıklığı çok artar. Bu gibi karmaşıklıkları ortadan kaldırmak için yeni Petri ağı türleri geliştirilmektedir.

Bu konuyu tanımlamak amacıyla, farklı özdeşlikler elde etmek için Petri ağlarında kullanılan jetonlara özel nitelikler veren ağlar önerilmiştir. Bu ağlar, yüklü geçiş ağlarını [Genrich ve Lautenbach, 1981], renkli Petri ağlarını [Jensen, 1981] ve kendine özgü jetonları olan yüksek seviyeli Petri ağları olarak ifade edilebilir [Reising, 1985]. Obje yöneltili (object oriented) Petri ağların [Sibertin-Blanc, 1985] ortaya konulması yüksek seviyeli Petri ağları için önemli bir gelişme olmuştur. Yaklaşık olan ve kesinlik taşımayan bilgileri ifade etmek amacıyla çeşitli Fuzzy Petri ağı [Chen ve Chang, 1990 – Grag ve ark., 1991 – Looney, 1988] modellerinin ortaya çıkmasına sebep olmuştur. Sistemlerin zamana bağlı analiz ihtiyacı için ise zamana bağlı Petri ağları ortaya çıkmıştır [Valette ve ark., 1989 – Suziki ve Lu, 1989].

Ayrık olay sistemlerinin kontrolü konusunda yapılan en önemli çalışma Ramadge ve Wonham tarafından ortaya konulan Gözetimli Kontrol Teorisi (Supervisory Control Theory-

SCT) olarak karşımıza çıkar [Ramadge ve Wonham, 1989]. Bu teori AOS'ler için kapalı çevrim denetleyicilerinin sentezlenmesine imkân sağlamıştır ve AOS'lerin tamamına uygulanabilen genel bir yöntemdir. AOS'lerin kontrolü için ortaya konan Gözetimli Kontrol Teorisi yöntemine alternatif olarak Petri ağı temelli yöntemler geliştirilmiştir [Guia ve DiCesare, 1991 - Sreenivas, 1994]. Bu yöntemlerin ortaya çıkmasının asıl nedeni Petri ağları kullanılarak geliştirilen yöntemlerin sonlu durum makinaları (SDM) kullanılarak geliştirilen yöntemlere göre çeşitli avantajlar sunuyor olmasıdır [Guia, 1996]. Öncelikle bir Petri ağının durumu o an ağıdaki jetonların dağılımı tarafından belirlenirken SDM'de sistemin içinde bulunduğu her farklı olasılığı ifade etmek için bir durum (state) kullanmak gerekmektedir. Böylece Petri ağının yapısı sistemdeki jeton sayısı artsa bile sisteme ait daha düzenli bir model ortaya koyabilmektedir. İkinci olarak, SDM'ler anlaması zor birçok karmaşık metinsel tanımlamalar ve matematik notasyonlarının kullanılmasını gerektirirken, Petri ağlarının kullanımında modellenen sistemi ve spesifikasyonları daha anlaşılır ağ yapıları şeklinde ifade etmek mümkündür. Aynı zamanda fiziksel sistemle elde edilen Petri ağı modeli arasında ilişki kurmak ve modeli anlamak çok daha kolaydır. Son olarak, Petri ağlarının kullanımı ile, aynı sistem modeline ait özelliklerin analiz edilmesi, performans gelişimi ve ayrık olay kontrolörlerinin sistematik tasarımı da yapılabilir.

Literatürde endüstriyel robotların Petri ağları kullanılarak modellenmesi ve kontrolü konusunda birçok çalışma bulunmaktadır. Moody ve Antsaklis tarafından yapılan çalışmada robotik montaj istasyonları için Petri ağları temelli geri beslemeli denetleyicilerin hesabı önerilmektedir. Bu çalışma da robotik montaj istasyonu için Petri ağı modeli tasarlanarak bu model üzerinde mevki değişmezleri tanımlanmaktadır. Robotik montaj istasyonu, çakışıklık (incidence) matrisi, mevki değişmezleri vektörü, başlangıç işaretleri ve sistem sınırlayıcıları oluşturularak sistem denetleyicileri kolay bir şekilde hesaplanmaktadır [Moody ve ark., 1995]. Filipescu ve arkadaşları tarafından yapılan bir diğer çalışmada da otonom bir endüstriyel robotik sistemin kontrolü çalışılmıştır [Filipescu ve ark., 2014]. Sistem robotik kol ve tekerlekli mobil robot olmak üzere iki aşama da parça kalite kontrolü için zamanlanmış hibrit Petri ağları modeli oluşturulmuştur. Oluşturulan Petri ağı modeli üzerinden sistem kontrolü karmaşıklığından uzak daha basit ve anlaşılır olmaktadır [Murata, 1989]. Petri ağı modeli çoklu robotlar ve görevler arasında programlanmış bir kontrol sağlamaktadır. Yöntem tüm robotların ve makinelerin eş zamanlı hareketini sağlar ve sağladığı senkronizasyon hareketlerinin komutları kullanıcı tarafından rahat anlaşılabilen robot dili olarak tanımlanmaktadır. Uygun robot dili ve koordinasyonla birden çok robot kolaylıkla

programlanarak kontrol edilebilmektedir [Yasuda, 2014 – Calouni ve ark., 1998]. Petri ağıları gibi otomata teorisi de endüstriyel robotların kontrolünde sıklıkla karşımıza çıkmaktadır. Otomata teorisi endüstriyel robotların kontrolünde işlem karmaşıklığını azaltarak çoklu robotların eş zamanlı çalışmalarında kolaylık sağlamaktadır [Rosell, 2004].

Morales ve arkadaşları tarafından yapılan çalışmada ise üretim sistemleri üzerindeki hatalı davranışlar ve bu hatalı davranışların çözümünde kullanılan Petri ağıları ve Bayes teoreminden bahsedilmiştir. Petri ağıları hatanın nelerden kaynaklandığını bulmada ve bu hataların hangi durumlarda nasıl ortadan kaldırılacağını ön görmede önemli katkı sağlamaktadır. Petri ağılarıyla sistem modellenerek olası hatalar kolaylıkla engellenebilmektedir. Bayes teoremiyle olası hataların ihtimalleri hesaplanmaktadır ve bu avantajlarla sistem uzaktan herhangi bir kördüğümüne sebebiyet vermeden kolaylıkla kontrol edilebilmektedir [Lee ve Lee, 2002]. Petri ağıları ile endüstriyel robot kontrolü dinamik ortamlarda karmaşaya sebebiyet vermeden çoklu robot çalışmalarına imkân sağlamaktadır [Morales ve ark. 2007 – Ziparo ve ark.1996].

Bu çalışma da günümüz endüstrisinde yaygın olarak kullanılmakta olan Robotik montaj istasyonlarının kontrolü ve programlanması üzerinde durulmuştur. Robot içeren bu sistemler Ayrık Olay Sistemi olarak sınıflandırılmakta olup kontrolü günümüzün güncel araştırma konularındandır. Tez kapsamında Gaziosmanpaşa Üniversitesi Mekatronik Mühendisliği bölümü Robotik ve Endüstriyel Otomasyon Laboratuvarında bulunan Mitsubishi RV- 2SDB robotunu içeren montaj istasyonu kullanılmıştır. Bu tez kapsamında ilk olarak robotik montaj istasyonu PA olarak modellenmiştir. Sistemin güvenli ve sorunsuz çalışması için gerekli olan kısıtlar kontrol spesifikasyonu olarak kabul edilmiş ve spesifikasyonları yerine getirecek denetleyici kontrol mevkileri ve ilgili geçişler olarak hesaplanmıştır. Elde edilen sistem modeli ve denetleyici yapısı PA formundadır. Sistem kontrolünü sağlayacak denetleyici yapısı montaj istasyonunu ve robotu kontrol eden denetleyici için gerçekleştirilmiştir. Gerçekleştirme işlemlerinde Melfa Basic IV programlama dili kullanılmıştır. Bu tez teorik denetleyici hesaplanması ve elde edilen denetleyicilerin gerçekleştirmesini bir araya getirmesi açısından literatüre katkı sağlamaktadır.

Bu tez beş bölümden oluşmaktadır. İlk bölüm olan giriş bölümünde tez konusundaki kaynak özetleri ve tezin önemi konularında bilgiler sunulmuştur. İkinci bölümünde kuramsal temeller olan Petri ağıları ve gözetimli kontrol açıklanmıştır. Üçüncü bölümde tez kapsamında kullanılan deney seti ve gözetici hesaplama yöntemine yer verilmiştir. Dördüncü bölümde sistem için elde edilen denetleyici yapısı ve bu yapının gerçekleştirilmesi sunulmuştur. Son bölümde ise sonuçlara yer verilmiştir.

2. KURAMSAL TEMELLER

Bu bölümde Petri Ağları (Petri Nets) ve Gözetimli Kontrol (Supervisory Control) hakkında tanıtıcı bilgiler sunulmaktadır. Daha detaylı bilgi [Lauzon, 1996] kaynaklarında bulunmaktadır.

2.1. Petri Ağları

Petri ağları 1939 yılında Alman matematikçisi Carl Adam Petri'nin doktora çalışmasında haberleşme uygulamaları için ağ (net) şeklinde tanımlanan yeni bir matematiksel araç olarak ortaya çıkmıştır. Temelde çeşitli sistemlerin modellenmesinde ve analizinde sıklıkla kullanılan grafiksel ve matematiksel bir araç olarak tanımlanan Petri ağları akış şemaları, blok diyagramları ve ağ yapılarına benzeyen modellerinden dolayı kullanıcıya görsel anlaşılabilirliği yüksek grafiksel modeller sunmaktadır [Murata, 1989]. Matematiksel araç olarak ise sistem davranışını cebirsel ifadeler kullanarak modelleme olanağı sağlamaktadır. Bu özellikleri sebebiyle Petri Ağları çoğunlukla, eş zamanlı, asenkron, dağıtık (distributed), belirsiz (nondeterminism) ve kaynak paylaşımı gibi karakteristik özellikleri içeren sistemlerin modellenmesinde etkin bir araç olarak kullanılmaktadır ve hem uygulama ağırlıklı çalışan araştırmacıları hem de teorik ağırlıklı çalışan araştırmacıları bir araya getirmektedir [David ve Alla 1994].

2.1.1. Basit Petri Ağları

Petri ağları (PA) matematiksel olarak basit bir dördlüyle ifade edilmektedir. Bu dördlü şu şekilde tanımlanmaktadır;

$$PA = (P, T, Pre, Post,) \quad (2. 1)$$

Burada, $P = \{ p_1, \dots, p_n \}$, sonlu mevkiler kümesini, $T = \{ t_1, \dots, t_n \}$, sonlu geçişler kümesini, Pre , P 'den T 'ye yönlenmiş oklar kümesine karşılık gelen $P \times T \rightarrow \{0,1\}$ giriş kodlamasını, $Post$, T 'den P 'ye yönlenmiş oklar kümesine karşılık gelen $T \times P \rightarrow \{0,1\}$ çıkış kodlamasını ifade etmektedir.

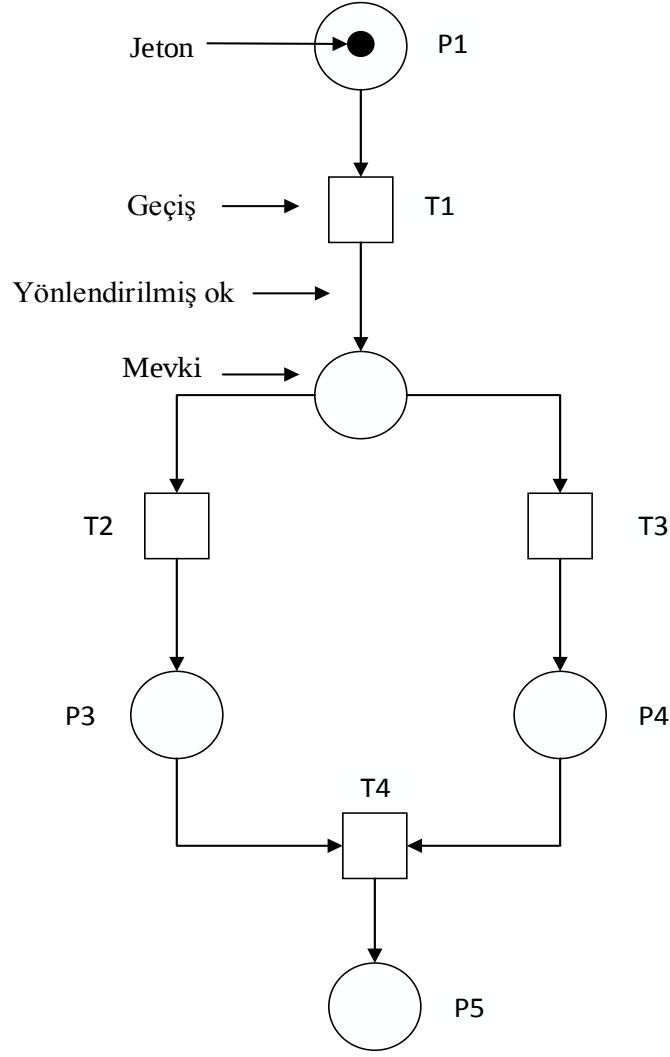
Burada P ve T ayrık kümelerdir ve $P \cup T$ 'nin herhangi bir elemanına *düğüm* denilmektedir. Petri ağlarının birbirine bağlı olduğu varsayıldığından herhangi iki düğüm arasında en az bir bağlantı var demektir. Geçişler sistemdeki bir durumdan diğerine olan kontrol gelişimine karşılık gelirken mevkiler, sistem durumlarını açıklamak için kullanılırlar.

Nasıl çalıştıklarının daha iyi anlaşılmasını sağlamak için Petri ağları grafiksel olarak gösterilebilirler. Petri ağı iki tip düğümden oluşur, dairelerle (O) gösterilen mevkiler ve karelerle (□) gösterilen geçişler. Giriş ve çıkış fonksiyonları iki düğüm arasındaki yönlendirilmiş oklarla ($\rightarrow$) gösterilirler. Bir mevkiden bir geçişe yönlendirilmiş olan ok geçişin giriş mevkisini, bir geçişten bir mevkisyeye yönlendirilmiş olan ok ise geçişin çıkış mevkisini göstermektedir [Gelen ve Uzam, 2012 – Gelen ve ark., 2007 – Uzam ve ark., 2007].

İşaretili bir Petri ağı mevkiler ve geçişlere ek olarak *jetonlar* içerir. PA'nın $M(P)$ işareti, her mevkideki jeton sayılarıyla gösterilmektedir, jetonlar siyah noktalarla (●) gösterilmektedir. Petri ağlarında sistem dinamiği ağ içerisindeki jetonların hareketi ile karakterize edilmektedir. Jetonlar mevkiler de bulunurlar ve oklar boyunca hareket etme yeteneğine sahiptirler. Jetonların ağdaki akışları ise geçişlerle kontrol edilmektedir. Jetonların mevkiler arasındaki hareketi PA'nın gelişimini gösterir ve bu, açık geçişlerin tetiklenmesiyle yapılır. Jetonların ağdaki akışı açılma ve tetikleme kuralları ile belirlenir. İşaretili basit bir Petri ağı aşağıdaki beşliyle ifade edilir.

$$PA = (P, T, Pre, Post, M) \quad (2.2)$$

M işareti i. bileşeni $M(p_i)$ olan n boyutlu bir vektör olarak gösterilir. $M(p_i)$ i. mevkideki jetonların sayısını gösterir. Şekil 2.1'de mevkileri, geçişleri, yönlü okları ve jetonu gösteren örnek bir Petri ağı modeli gösterilmektedir.

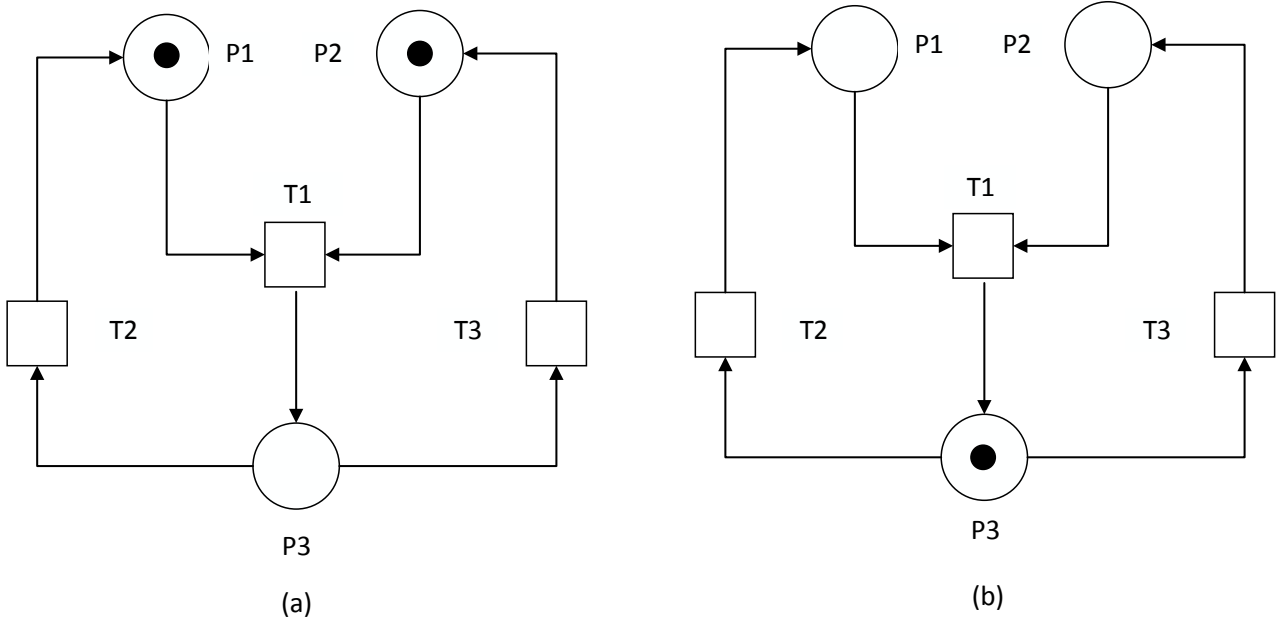


Şekil 2.1. Basit bir Petri ağı modeli

Basit bir Petri ağının çalışması, jetonların ağdaki sayısı ve jetonların ağ içerisine dağılımıyla kontrol edilir ve jetonların ağ içerisinde akmasına sebep olur. Petri ağının çalışması açık geçişlerin tetiklenmesiyle ve tetiklenen bu geçişlerden jetonların akmasıyla gerçekleşir. Bir geçiş, giriş mevkilerinin her birinin en az bir jetonla işaretlendiği zaman aktif olur ve bu durumda geçiş *açıktır*. Bir geçiş, geçişe bağlı olan giriş mevkilerinin her birinden bir jeton alınıp, çıkış mevkilerinin her birine bir jeton yerleştirmesiyle *tetiklenir*. Geçişlerin tetiklenmesi jetonların Petri ağı içerisinde akmasına sebep olur.

2.1.2. Basit Bir Petri Ağının Tetiklenmesi

Basit bir Petri ağının tetiklenmesi Şekil 2.2’de görülmektedir. Burada üç mevki, $P = \{p_1, p_2, p_3\}$ ve üç geçiş, $T = \{t_1, t_2, t_3\}$ vardır. İlk olarak Şekil 2.2(a)’da görüldüğü gibi t_1 geçişi açıktır, çünkü $M(p_1) = 1$, $Pre(p_1, t_1) = 1$ ve $M(p_2) = 1$, $Pre(p_2, t_1) = 1$ ’dir. t_2 ve t_3 geçişleri açık değildir. Çünkü $M(p_3) = 0$, $Pre(p_3, t_2) = 1$ ve $Pre(p_3, t_3) = 1$ ’dir. t_1 geçişi tetiklendiğinde p_1 ve p_2 mevkilerinden birer jeton alınır ve Şekil 2.2(b)’de görüldüğü gibi p_3 mevkisine bir jeton yerleştirilir. Bu durumda t_2 ve t_3 geçişleri açılır, çünkü $M(p_3) = 1$, $Pre(p_3, t_2) = 1$ ve $Pre(p_3, t_3) = 1$ ’dir. t_2 veya t_3 geçişlerinden herhangi birinin tetiklenmesiyle p_3 mevkisinden bir jeton alınır ve Şekil 2.2(a) p_1 ve p_2 mevkilerine bir jeton yerleştirilir.



Şekil 2.2. Basit bir Petri ağında : (a) ilk işaretleme. (b) t_1 ’in tetiklenmesinden sonraki işaretleme

2.1.3. Petri Ağlarının Özellikleri

Petri ağları kullanılarak modellenmiş bir sistemin davranışsal ve yapısal özellikleri tespit edilebilir. Davranışsal özellikler Petri ağının ilk işaretine bağlıken, yapısal özellikler ilk işarete bağlı değildir ve verilen Petri ağının yapısına bağlıdır. Bu kısımda bazı davranışsal özellikler açıklanmaktadır.

Erişilebilirlik; açık bir geçişin tetiklenmesiyle meydana gelen jeton akışının mevki işaretinin değişmesine sebep olmasıdır.

Sınırlılık; her bir mevkide ki jetonların sayısı ilk işaret M_0 ' dan erişilebilir olan sistemdeki her işaret için sonlu bir sayı olan n sayısını geçmiyorsa bu Petri ağı modeli n -sınırlı yada *sınırlıdır* denir.

Kararlılık; bir Petri ağının bütün mevkilerinde ki jeton sayılarının toplamı, sistemdeki bütün işaretler için sabitse bu Petri ağı *kararlıdır* denir.

Güvenilirlik; eğer bir p mevkisi birden fazla jeton içermiyorsa ya da başka bir deyişle Petri ağı 1 ile sınırlıysa güvenilirdir. Petri ağında ki bütün mevkiler güvenilir ise Petri ağı *güvenilirdir* denir.

Canlılık; bir geçiş için, Petri ağının bütün işaretlerinde, ağı ilk işarete götüren bir dizi tetikleme varsa o geçiş canlıdır ve Petri ağının bütün geçişleri canlı ise o Petri ağı *canlıdır* denir.

Tersinirlik; ilk işaret M_0 'a diğer bütün işaretlerden erişilebiliyorsa o Petri ağı *tersinirdir* denir [Zhou ve Venkatesh, 1999 – Belami ve ark., 1993 – Moody ve ark., 1998].

2.1.4. Petri Ağlarının Cebirsel Özellikleri

2.1.4.1. Petri Ağlarında Çakışıklık Matrisi (Incidence Matris)

İşaretili bir Petri Ağı (PA) $R = \{Q, m_0\}$ Q işaretsiz bir PA, m_0 başlangıç işaretidir [David ve Alla, 2005]. Çakışıklık matrisi C_N , mevkilerin satırlara, geçişlerin ise sütunlara karşılık gelecek şekilde yazılmasıyla oluşturulan matristir. Her bir geçiş sütunu ağ üzerindeki işaretlemenin geçiş üzerindeki etkisini temsil etmektedir.

$$C_N = P \times T \rightarrow Z \quad (2.3)$$

$$C(t, p) = W(t, p) - W(p, t) \quad (2.4)$$

Giriş çakışıklık matrisi (input incidence matris) ;

$$W(p, t) = W_{ij}^- = [w_{ij}^-] \quad , \quad w_{ij}^- = \text{Pre}(p_i, t_j) \quad (2.5)$$

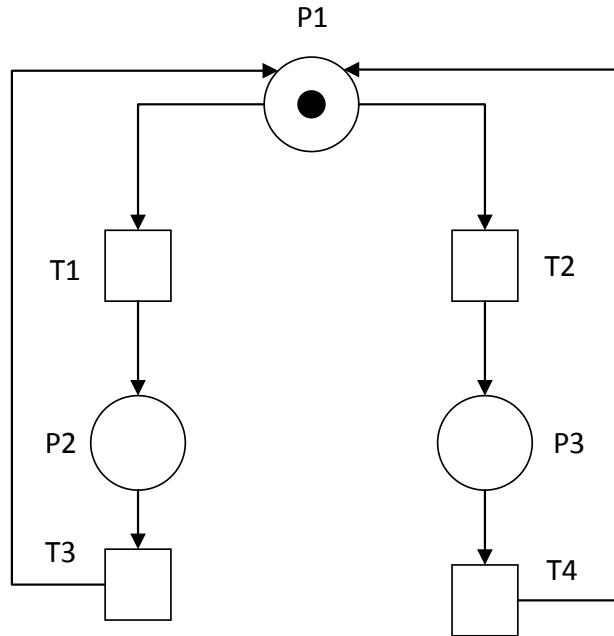
Çıkış çakışıklık matrisi (output incidence matrix) ;

$$W(t, p) = W_{ij}^+ = [w_{ij}^+] \quad , \quad w_{ij}^+ = \text{Post}(p_i, t_j) \quad (2.6)$$

olarak adlandırılır. Şekil 2.3’de örnek bir Petri ağı modeli görülmektedir. Bu ağ için giriş ve çıkış çakışıklık matrisleri şu şekilde hesaplanmaktadır.

$$W^- = \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & T_3 & T_4 \end{array} \\ \begin{array}{l} \left[\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right] \begin{array}{l} P_1 \\ P_2 \\ P_3 \end{array} \end{array} \quad W^+ = \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & T_3 & T_4 \end{array} \\ \begin{array}{l} \left[\begin{array}{cccc} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array} \right] \begin{array}{l} P_1 \\ P_2 \\ P_3 \end{array} \end{array}$$

$$W = W^+ - W^- = \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & T_3 & T_4 \end{array} \\ \begin{array}{l} \left[\begin{array}{cccc} -1 & -1 & 1 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{array} \right] \begin{array}{l} P_1 \\ P_2 \\ P_3 \end{array} \end{array}$$



Şekil 2.3. Örnek Petri ağı modeli

2.1.4.2. Petri Ağlarında İşaretleme Vektörleri

Petri ağlarında işaretleme vektörleri sütun vektörü olarak yazılmaktadır ve başlangıç işaretleme vektörü, başlangıçta içerisinde jeton bulunan mevkileri göstermektedir. Şekil 2.3'deki örnek petri ağı modelinin başlangıç işaretleme matrisi şu şekildedir;

$$M_0 = [1 \ 0 \ 0]^T$$

Eğer t_1 ve t_2 geçişleri tetiklenirse elde edilecek yeni işaretleme vektörü,

$$u = [1 \ 1 \ 0 \ 0]^T \text{ olur.}$$

Bu geçişlerin tetiklenmesi sonucunda oluşan yeni işaretleme vektörü şu şekilde hesaplanmaktadır;

$$M = M_0 + C \cdot u \quad (2.7)$$

$$M = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & -1 & +1 & +1 \\ +1 & 0 & -1 & 0 \\ 0 & +1 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ +1 \\ +1 \end{bmatrix}$$

2.1.4.3. Petri Ağlarında Mevki ve Geçiş Değişmezleri

Petri Ağları (PA)'nda P mevkileri temsil etmektedir. İşaretleme bir değişmez (marking invariant) lineer mevki değişmezi ya da basit olarak değişmez olarak adlandırılmaktadır. Mevki değişmezleri P' nin içerdiği $P' = \{P_1, P_2, \dots, P_r\}$ alt kümelerinden oluşmaktadırlar. Ağırlık vektörleri ise $(q_1, q_2, \dots, q_r)$ bütün ağırlıklar için q_i pozitif tam sayılardır. Şu şekilde gösterilir [38];

$$q_1 \cdot mP_1 + q_2 \cdot mP_2 + \dots + q_r \cdot mP_r = \text{sabit} \quad \text{her } m \in M(m_0) \quad (2.8)$$

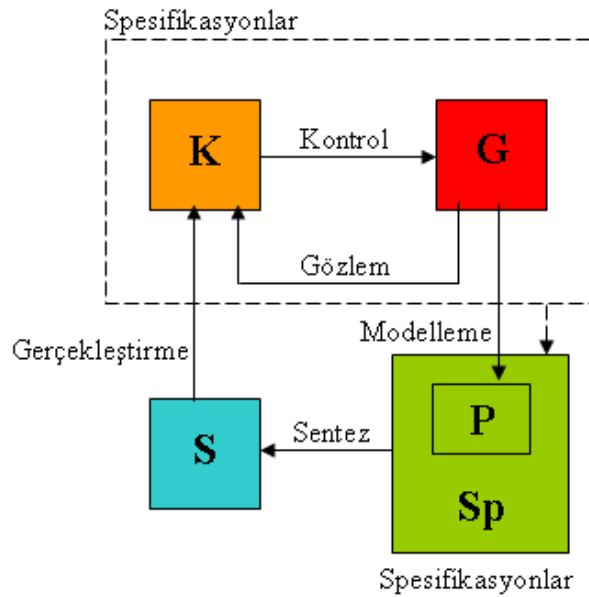
$M = M_0 + C \cdot u$ eşitliğinin çözümünde $C \cdot u = 0$ olduğu durumlar geçiş değişmezi (T-invariants) olarak adlandırılmaktadır.

$M = M_0 + C \cdot u$ eşitliğinin çözümünde $C^T x = 0$ olduğu durumlar mevki değişmezi (P-invariants) olarak adlandırılmaktadır. Burada x (2.8) eşitliğinden çıkarılan mevki değişmezi vektörüdür.

Mevki değişmezleri bazı lineer değişmezlerin bütün ulaşılabilir işaretlemlerindeki jeton sayılarını göstermektedir. Şekil 2.3.'teki örnek petri ağı modeli üzerinden düşünülecek olursa p_1 ve p_2 mevkilerinde aynı anda birden fazla jeton bulunamaz. Yani $\mu_1 + \mu_2 \leq 1$ olmaktadır.

2.2. Gözetimli Kontrol (Supervisory Control)

Gerçek bir sistemin kontrolü için gerekli adımlar Şekil 2.4.'te gösterilmektedir. Bu aşamalarda modelleme ile G gerçek sisteminin P isimli modeli oluşturulmaktadır. G gerçek sisteminin K gerçek denetleyicisi ile oluşturduğu kapalı çevrim sisteminin özellikleri Sp (spesifikasyon) olarak modele aktarılmaktadır. P ve Sp 'yi birlikte dikkate alarak bir S kontrolörünün oluşturulması işleme sentez denilmektedir. Sentezlenmiş bir S denetleyicisinin karşılığı olan donanım yapısına ise gerçekleştirme denilmektedir.



Şekil 2.4. Kontrol aşamaları [Gelen ve Uzam, 2012]

Tez kapsamında Festo tarafından üretilmiş olan ve içerisinde Mitsubishi RV- 2SDB robotu bulunan robotik montaj istasyonu Petri ağları ile modellenmiştir. Moody ve Antsaklis tarafından önerilen mevki değişmezleri olarak ifade edilen yöntem kullanılarak kontrol spesifikasyonlarına ilişkin denetleyici hesaplanmıştır. Tezin ileriki kısımlarında bu hesaplanan denetleyicinin gerçekleştirilmesi açıklanmaktadır.

3. MATERİYAL VE YÖNTEM

Bu bölümde tez kapsamında kullanılan Mevki Değişmezleri ile Gözetleyici Sentezlenmesi, Mitsubishi RV – 2SDB Robotunun Genel Özellikleri, Robot Montaj İstasyonu ve Melfa Basic robot programlama dili tanıtılmaktadır.

3.1. Mevki Değişmezleri Kullanılarak Gözetici Sentezlenmesi

Moody ve Antsaklis [Moody ve Antsaklis, 1998] tarafından önerilmiş olan Mevki değişmezleri kullanılarak gözetici hesabı metodu kullanılarak PA olarak modellenmiş sistemler için gözetici hesabı yapılabilmektedir. Bu gözetici hesabında öncelikle sistem Petri ağı modeli üzerinden D_p çakışıklık matrisi oluşturulmaktadır sonrasında sistem kısıtları belirlenir. Bu sistem kısıtlarına ilişkin denklem oluşturulur ve gözetici hesabı yapılır.

Petri ağları üzerinde mevki değişmezleri kullanılarak onunla ilgili her bir ağın değişmeziyle ilişkilendirilerek yeni bir gözetleyici hesaplanabilmektedir. Her bir kısıt kontrolü için D çakışıklık (incidence) matrisi bir satır daha arttırılır. Böylelikle D iki matrisin birleşiminden oluşur. D_p sistemin asıl matrisidir ve D_c sistemin kontrol matrisidir. Burada X vektörü (2.8) eşitliğinden çıkarılan mevki değişmezi vektörüdür.

$$X^T = [l_1, l_2, \dots, l_n, 1] \quad (3.1)$$

Bu kontrol problemi aşağıdaki adımların takibiyle çözümlenmektedir. Bütün sınırlayıcılar gruplanabilir ve matris formunda yazılabilir.

$$L \mu_p \leq b \quad (3.2)$$

μ_p sistemin işaretleme vektörüdür, $L \in Z^{(n+n_c)}$, $b \in Z^{n_c}$ ve n_c kısıtlamaların miktarıdır.

Eşitsizliğin $L \mu_p$ ve b gibi vektörel elementlerin ayrık lojik birleşimiyle oluştuğu düşünülebilir [33]. Basit değişkenler içeren kısıtlamalar,

$$L \mu_p + \mu_c = b \quad (3.3)$$

$\mu_c \in Z^{n_c}$ Kontrol yerlerinin işaretlemesini yansıtan tamsayı vektörüdür.

D_c matrisi sistemin geçişleriyle kontrol yerlerinin bağlantılarını içerir. D çakışıklık (incidence) matrisi $D \in Z^{(n+n_c) \times m}$, kapalı çevrim sistemi şu matris tarafından verilir.

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix} \quad (3.4)$$

ve işaretleme vektörü $\mu \in Z^{(n+n_c)}$ ve başlangıç işaretlemesi μ_0 dır.

$$\mu = \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix} \quad \mu_0 = \begin{bmatrix} \mu_{p0} \\ \mu_{c0} \end{bmatrix} \quad (3.5)$$

Bu denklemlerden şu eşitliği çıkarabiliriz.

$$X^T D = \begin{bmatrix} L & I \end{bmatrix} \begin{bmatrix} D_p \\ D_c \end{bmatrix} = 0 \quad (3.6)$$

$$LD_p + D_c = 0 \quad (3.7)$$

Burada $I \in Z^{(n+n_c)}$ (3.1) denkleminde bütün eşitlikler 1'e eşitlendiğinde uzay değişkenlerinin katsayı matrisi oluşmaktadır.

$$b - L \mu_{p0} \geq 0 \quad (3.8)$$

μ_{c0} başlangıç işaretlemesiyle, $D_c \in Z^{(n_c \times m)}$ bir Petri ağı denetleyicisidir,

$$D_c = -LD_p \quad (3.9)$$

$$\mu_{c0} = b - L \mu_{p0} \quad (3.10)$$

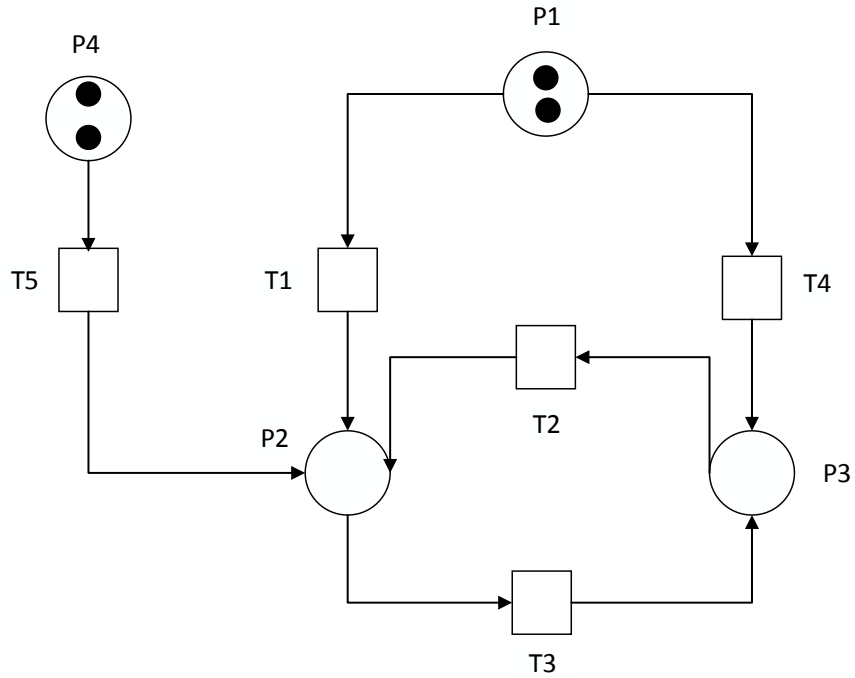
Eşitlik (3.6)'nın eşitlik (3.4)'e göre çözümü yapılırsa $[L \ I]$ kuvvetleri kapalı çevrim sisteminin değişmezleridir. $[L \ I]$ matrislerinin değişmeyeceğini bildiğimize göre $L \mu_p + \mu_c = L \mu_{p0} + \mu_{c0}$ olmaktadır. Eşitlik (3.7)'de $L \mu_{p0} + \mu_{c0} = b$ olduğuna göre, bu yer işaretlemesi genellikle sıfıra eşittir ya da sıfırdan büyüktür.

$$L \mu_p + \mu_c = b \quad (3.11)$$

$$L \mu_p \leq b \quad (3.12)$$

olmaktadır.

Mevki değişmezleri ile gözetleyici sentezlenmesine ilişkin Şekil 3.1'de verilmiş olan örnek Petri ağı modelinin D_p çakışıklık matrisi, μ_{p0} başlangıç işaretlemesi, D_c kontrolör matrisi ve hesaplamaları aşağıda verilmektedir.



Şekil 3.1. Mevki değişmezleri kullanılarak gözetleyicisi hesaplanacak örnek Petri ağı modeli [39]

Petri ağı modelinin çakışıklık (incidence) matrisi mevki ve geçiş durumları incelenerek şu şekilde bulunmuştur.

$$D_p = \begin{bmatrix} -1 & 0 & 0 & +1 & 0 \\ +1 & +1 & -1 & 0 & +1 \\ 0 & -1 & +1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

μ_{p0} , başlangıç işaretlemesi şu şekildedir;

$$\mu_{p0} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

$X^T \cdot D_p = 0$ olduğundan

$X^T = [1 \ 1 \ 1 \ 1]$ olmaktadır.

Kontrol spesifikasyonu p_2 ve p_3 mevkilerinde toplam en fazla 1 adet jeton olmasıdır. Bu kısıt aşağıdaki eşitsizlikle ifade edilebilir;

$$\mu_2 + \mu_3 \leq 1$$

$$b \leq 1$$

Bu eşitlikten elde edilecek işaretleme şu şekildedir;

$$L = [0 \ 1 \ 1 \ 0]$$

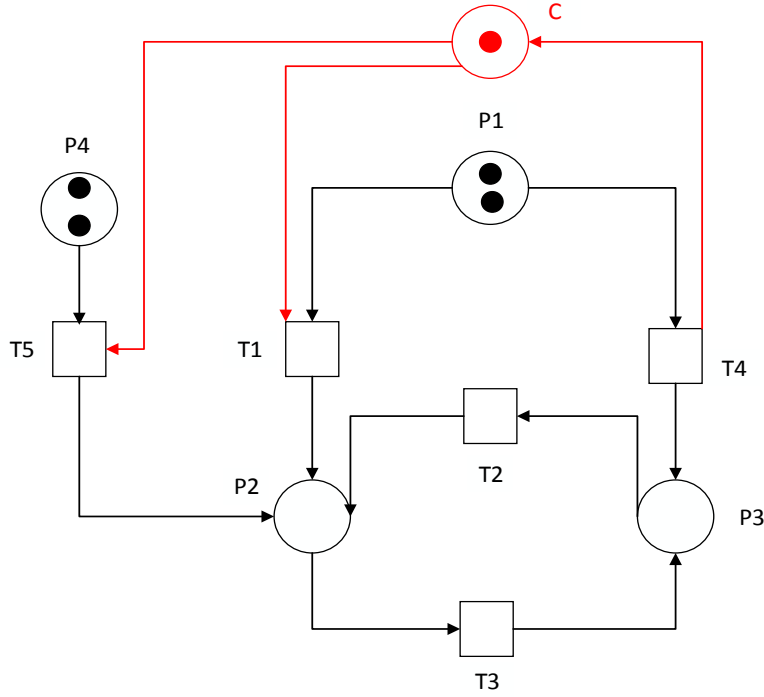
$$\mu_{c0} = b - L \mu_{p0}$$

$$\mu_{c0} = 1 - [0 \ 1 \ 1 \ 0] \cdot \begin{bmatrix} 2 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

$$\mu_{c0} = 1 - 0 = 1$$

$$D_c = -L \cdot D_p = [-1 \ 0 \ 0 \ +1 \ -1]$$

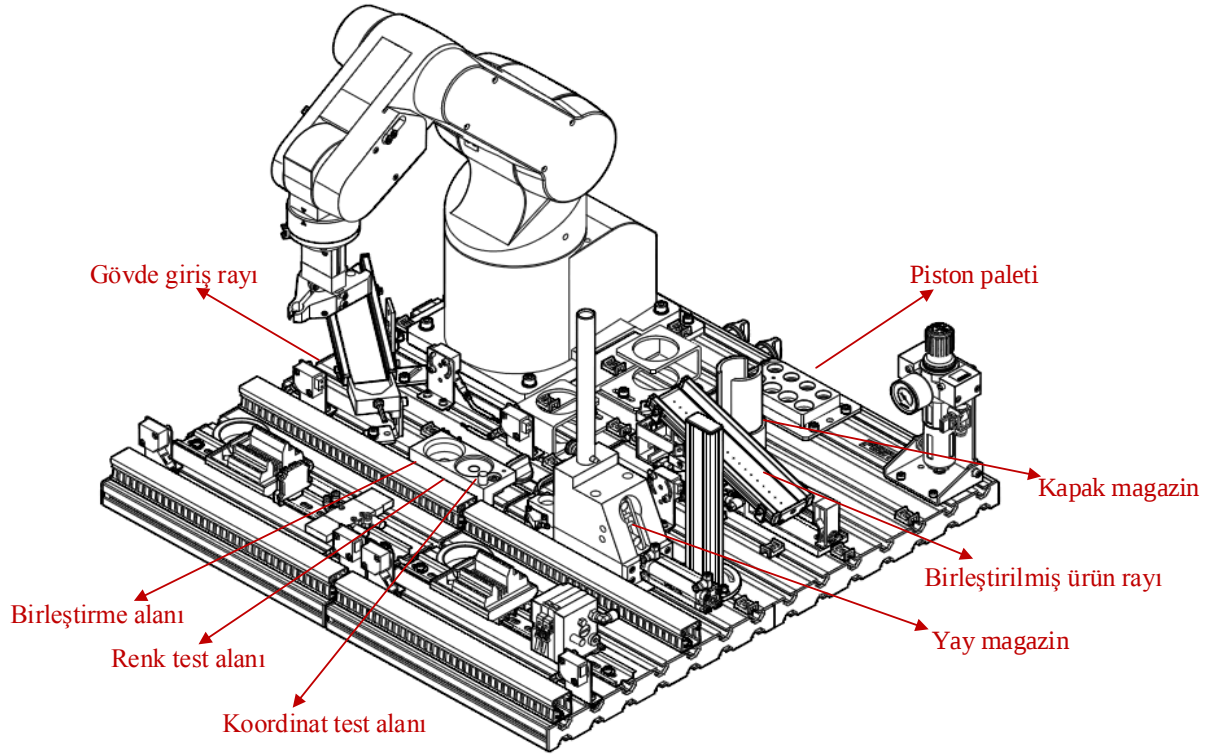
Bu eşitlikten elde edilen C kontrol mevkisi Şekil 3.2’de gösterilmektedir [39].



Şekil 3.2. Mevki değişmezleri kullanılarak gözeticisi hesaplanan Petri ağı modeli ve C kontrolörü [David ve Alla, 2005]

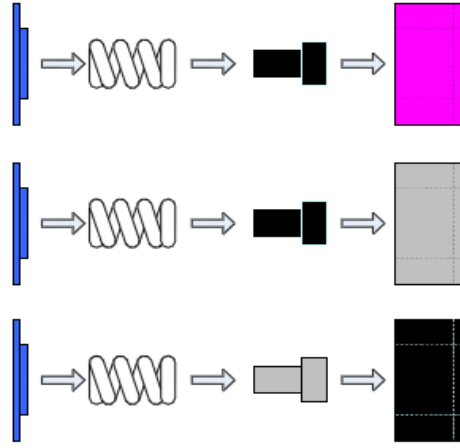
3.2. Robot Montaj İstasyonu

Festo tarafından üretilen robotik montaj istasyonu bünyesinde Mitsubishi tarafından üretilmiş olan RV- 2SDB robotunu barındırmaktadır. Kullanılacak deney düzeneğinin resmi Şekil 3.3’te görülmektedir. Bu deney düzeneğinde kırmızı, metal ve siyah olmak üzere üç farklı renkte silindir gövdesi bulunmaktadır. Kırmızı ve metal silindir gövdeler içerisine siyah piston, siyah silindir gövdesi içerisine metal piston yerleştirildikten sonra yay yerleştirilerek plastik bir kapak kapatılmak suretiyle örnek bir ürünün montajı Robot ve diğer endüstriyel sistemler vasıtasıyla yapılabilir. Sistemde montajı yapılabilen parçalar Şekil 3.4’te görülmektedir [Anonim, 2015].



Şekil 3.3. Deney düzeneğinin resmi

Deney düzeneğinin çalışma prensibi şu şekildedir; öncelikle gövde giriş rayına bir tane silindir bırakılır. Silindirin geldiği sensör tarafından algılanır ve robot kolu hareket geçerek silindir alınır. Alınan silindir robot kolu tarafından renk testi alanına götürülüp bırakılır. Daha sonra robot kolu silindirin yanında 1 cm boşluk kalacak şekilde silindirin yanına yaklaşarak rengini algıladıktan sonra yanından kalkar ve tekrar silindiri alıp koordinat test alanına götürerek orda alt kısmında ki boşluğu buluna kadar döndürür ve doğru açığı yakaladığında silindiri birleştirme kısmına götürerek yerine yerleştirir. Silindirin rengi siyah ise metal renkli pistonu, silindirin rengi kırmızı ya da metal ise siyah renkli pistonu alarak silindirin içine yerleştirir. Piston yerleştirildikten sonra robot kolu yayı alarak pistonun üzerine yerleştirir. Sonrasında robot kolu plastik kapağı koordinat testi alanına getirerek silindirin üzerine doğru açıda yerleştirebilmek için koordinat testini yaptıktan sonra silindirin üzerine yerleştirerek montaj işlemi tamamlanmış olan pistonu birleştirilmiş ürün rayı kısmına bırakır ve montaj işlemi tamamlanmış olur.

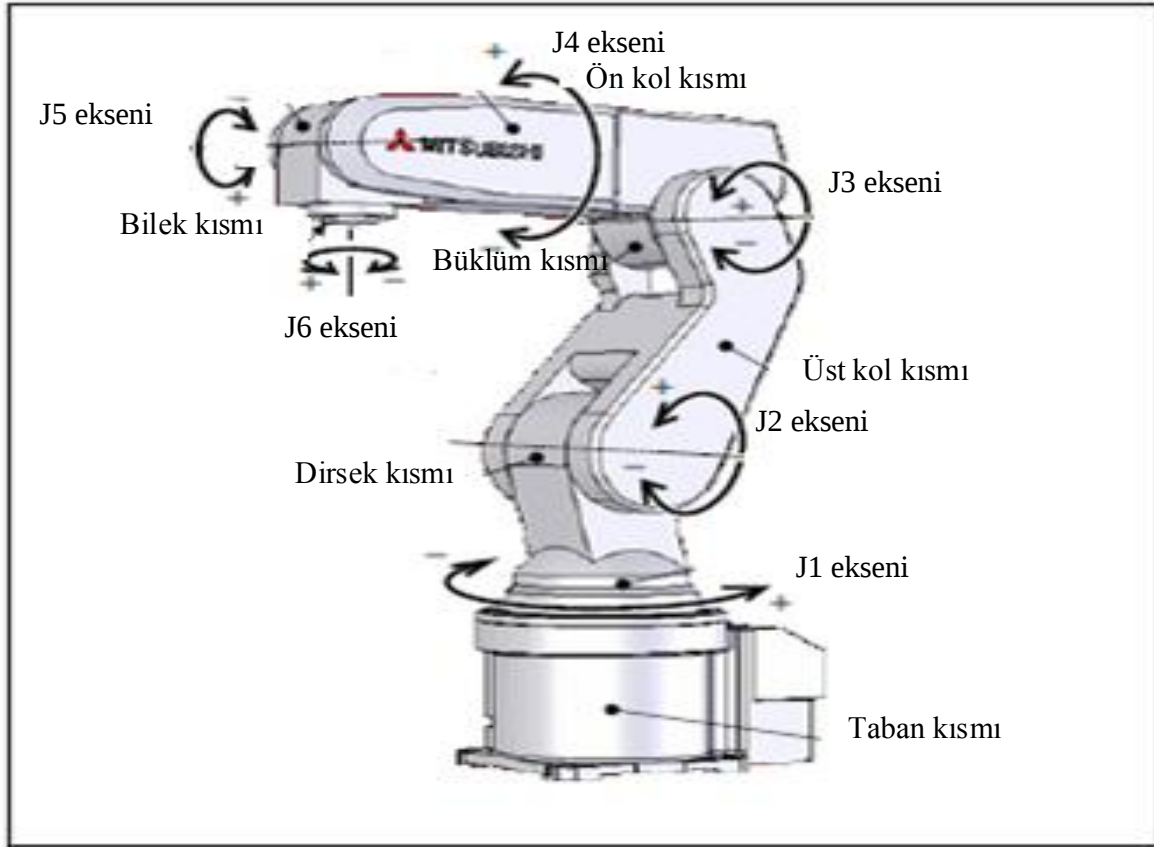


Şekil 3.4. Deney düzeneği ile montajı yapılan parçalar

Deney düzeneğinin kontrolü robot montaj istasyonu üzerinde bulunan CR1- 571 kontrolörü ile sağlanmaktadır. CR1- 571 kontrolörü 64 bitlik RISC (Reduced instruction set computing- indirgenmiş komut takımı mimarisi) ve DSP (Digital signal Processing- sayısal işaret işleme) mimarisine sahiptir. 2500 nokta 5000 adım ve 88 adet program hafıza kapasitesine sahiptir. Robot programlama dili olarak Melfa Basic IV Programlama dili kullanılmaktadır. Harici 16 giriş ve harici 16 çıkış portuna sahiptir.

3.3. Mitsubishi RV- 2 SDB Robotunun Genel Özellikleri

Mitsubishi tarafından üretilmiş RV – 2 SDB Robotunun blok resmi Şekil 3.5’te verilmiştir. Robot 6 eksenli endüstriyel montaj robotu olup robotun ana kısımları ve eksenleri şu şekildedir. Robotun üzerine oturduğu taban kısmı, J1 eksenini etrafında dönmeyi sağlayan orta kısım, J2 eksenini etrafında dönme işlemini gerçekleştiren omuz kısmı, J2 eksenini ile J3 eksenini arasında bulunan üst kol kısım. J3 eksenini etrafında dönme işlemini gerçekleştiren dirsek kısmı, J3 eksenini ile J4 eksenini arasında bulunan büküm kısmı. J4 eklemi ile J5 eklemi arasında bulunan ön kol ve J5 eklemi ile J6 eklemi arasında ki bilek kısmından oluşmaktadır [Anonim, 2015].



Şekil 3.5. RV – 2 SDB Robotu

3.4. Melfa Basic Programlama Dili

Bütün Mitsubishi robotlarıyla kullanılabilen PC tabanlı, güçlü bir programlama yazılımıdır. Programlama, denetim ve hata ayıklama bu yazılım paketinin özelliklerinden sadece birkaçıdır.

Melfa Works robotik yazılım programı olarak robot ve robot hücrelerinin programlanması ve simülasyonu için kullanabileceğiniz üç boyutlu bir yazılımdır. Gerçek uygulamalarda kullanmaya başlamadan önce, akış programında hata kontrolü yapılabilmektedir [Mitsubishielectric].

Bu tez de Melfa Basic IV programlama dili kullanılmakta olup çalışmalar Cosimir programı üzerinden gerçekleştirilmiştir. Cosimir robotik araştırma ve projelerinde kullanılan taslaktan kurulumu kadar birçok aşamanın üzerinde gerçekleştirilebildiği esnek bir yapıya sahip yazılım dilidir.

3.4.1. Robot Hareketleri

Bütün robot programlarının temel amacı robotu hareket ettirerek montaj, taşıma, kaynak gibi işleri yapmaktır. Bu sebeple MELFA BASIC deki hareket komutları çok önemlidir. En çok bilinen ve kullanılan hareket tipleri lineer hareket (lineer enterpolasyon) ve noktadan noktaya hareket (eksen enterpolasyonu) dir. Pozisyonlar robotun hareket hattını oluşturan noktaları tanımlamaktadır. Pozisyon listesinde pozisyon isimleri ile tanımlıdır. Pozisyon listesi; Öğretme (teach-in) metodu, tutucunun pozisyon bilgilerini edinmede ve robota öğretme de en çok bilinen ve kullanılan metottur. Veriler pozisyon listesinde saklanır. Her programın kendi için özel pozisyon listesi vardır. Program ve pozisyon listeleri isimleriyle atanırlar ve çağırılırlar. Program içerisinde kullanılan pozisyon isimleri pozisyon listesinde bulunmalıdır. Pozisyon isimleri program içerisinde pozisyon değişkenlerine değerlerin atanmasında kullanılmaktadır.

3.4.1.1.Lineer Hareket

Takım merkezinin önceden tanımlanmış bir doğrusal yörünge üzerinde hareket ettirilmesi sıklıkla kullanılan bir yöntemdir. Çoğu montaj, taşıma vb. operasyonlar robotun doğrusal yörüngede hareketini gerektirir. Doğrusal hareketler iki nokta ile tanımlanır. Programlamada hedef pozisyonun belirtilmesi doğrusal hareket için yeterlidir. Karmaşık yörüngelerde arada üçüncü, dördüncü pozisyonlar tanımlanmalıdır. Robot bu ara pozisyonlardan durmadan geçer.

3.4.1.2.Noktadan Noktaya Hareket

Parçaların bir P1 noktasından P2 noktasına hareket ettirilmesidir. Lineer harekette olduğu robotun hiçbir yere çarpmadan hareketini sağlamak için ara pozisyonlar tanımlanmalıdır. Bu ara pozisyonlarda robot durmadan hareketine devam edecektir.

3.4.2. Hareket Komutları

Bu kısımda bazı hareket komutları ve görevleri kısaca tanıtılmaktadır.

HOPEN ve HCLOSE (tutucu aç kapa): Robot elinin açılıp kapanmasını kumanda eden komutlardır.

MOV: Belirtilen pozisyona eklem enterpolasyonu ile gider.

MVS(hareket et): Belirtilen pozisyona doğrusal enterpolasyonla gider.

MVC: Yörüngesi başlangıç noktası, geçiş noktası ve bitiş noktasıyla tanımlanan 3 boyutlu dairesel enterpolasyon yapar. (Dairesel enterpolasyonla hareket edilen yörünge 3 nokta ile tanımlan kapalı bir dairedir (360 deg.)). Dairesel enterpolasyonun başlama noktası o anki pozisyon değilse, komut işlediğinde robot otomatik olarak doğrusal enterpolasyonla başlangıç noktasına gider. Komut işletilirken program durdurulur, robotun pozisyonu el ile değiştirilir ve sonra program tekrar başlatılırsa robot kaldığı yere doğrusal enterpolasyonla gider.

CNT (süreklilik): Hareketler sırasında başlangıç ve bitiş noktaları arasında tanımlanan noktalarda robot durmadan hareket eder. Robotun çizdiği yörüngenin keskin noktalarla kırılmaması için yuvarlatma yapılır. Verilen mesafe değerleri robotun asıl noktaya kaç mm kala yuvarlatmaya başlayacağını ve yuvarlatmayı bitireceğini gösterir. Bu değerlerin büyük seçilmesi robotun yörüngesinden sapma miktarını artırır.

BASE: Temel koordinatı değiştirmek için kullanılır. Robotun hareketlerinde referans aldığı temel noktadır. Komutla yeni tanımlanan noktanın eski noktaya göre X, Y ve Z eksenlerindeki konumları yazılır. A, B, C ile dik koordinat eksenlerine açılı bir tanımlama yapılabilir. Açılar saat ibresi yönünde pozitifdir. Temel koordinatın fabrika değeri P_NBASE durum değişkeninde kayıtlıdır. Temel koordinat tekrar bu noktaya getirilmek isteniyorsa P_NASE sistem değişkeninden çağrılır.

SPD(hız): Doğrusal ve dairesel harekette hız mm/sn cinsinden tanımlanır.

ACCEL (ivmelenme): Robotun hızlanma ve yavaşlama sürelerini ayarlar. İvmelenme oranı düşürüldükçe hızlanma ve yavaşlama için geçen zaman da artar.

3.4.3. Tanımlama Komutları

DEF JNT (eklem değişkeni tanımlama): Eklem değişkeni tanımlar. Değişken en fazla 8 karakter olmalıdır. Özel karakter kullanılmamalıdır.

DEF PLT (palet tanımla): Bütün pozisyonlarının kontrol tarafından hesaplandığı bir palet tanımlar.

DEF POS (Pozisyon değişkeni tanımlama): Programda Pozisyon olarak algılanacak herhangi bir isim altında değişkenler oluşturur.

Değişkenin ismi en fazla 8 karakter olmalıdır. Kullanılan değişken *character* veri tipinde olmalıdır. Birden çok değişkeni bir komut satırında tanımlarken komut satırının 123 karakteri geçmemesine dikkat edilmelidir. Tanımlanan değişken *pozisyon değişkeni* kurallarına uygun olmalıdır. Tanımlandıktan sonra değişken tıpkı *P* değişkeni gibi kullanılabilir.

3.4.4. Haberleşme Komutları

MELFA-BASIC I/O haberleşmesi için özel komutlar içermez. Bunun yerine robot durum değişkenlerini kullanır. M_IN(kanal) ve M_OUT(kanal) durum değişkenleri 1 ile 32767 arasında değerler alabilir. Bu değişkenlerin sayısını kontrolördeki I/O kartının giriş sayısı sınırlar.

OPEN: Dosya yada haberleşme kanalını açar.

FILE tanımlanırken adı 16 karakteri aşmamalıdır. Erişim şekli *INPUT*, *PRINT* ve *APPEND*’dir. Haberleşme hattı dosyası tanımlanırken dosyalar COM1:, COM2:, COM3: ismini alır. Erişim şekli *Omitted*’dir. Haberleşme hattından kesme yapılacaksa dosya numaraları 1 ile 3 arasında olur. Kesme yoksa 1’den 8’e kadar olabilir. COM1 dosyası standart RS-232-C’ yi ifade eder. COM2 ve COM3 de tanımlanabilir.

CLOSE: Belirtilen dosyayı kapatır.

INPUT: Bir dosyadan veya sürücüden veri alır.

PRINT: Bir dosyaya veri yazdır.

3.4.5. Çoklu iş (multitasking) komutları

Kontrol operasyonlarında birkaç programın aynı anda çalışabilmesi avantajını sağlar. Bu nedenle MELFA-BASIC her kontrol çevriminde bütün süreçlerin hesaplanmasını sağlayan time slot mekanizmalarını destekler. Paralel süreç için başlatma, yükleme, yönetim komutları aşağıda görülmektedir.

GETM: Robotu kontrol altına alma.

XLOAD: Programı seç ve belirtilen çoklu-iş yuvasına yükle.

XRUN: Çoklu-iş yuvasında programı çalıştırır. Bu komuttan önce XLOAD komutu ile program yuvaya yüklenmelidir.

XSTP: Önceden başlatılan işlemi durdurur.

XRST: Belirtilen yuvada işlemekte olan program durdurulur ve başa alınır.

3.4.6. Programlama Komutları

REM(yorum ve açıklama): Kullanıcı tarafından girilebilen en fazla bir program satırı uzunluğunda karakterlerden oluşan açıklama metni için kullanılır. Bu komutun ardından yazılanlar program tarafından işleme alınmaz. Programcayı aydınlatıcı bilgileri içerir. REM yerine (') kesme işareti de kullanılabilir.

DIM (bir dizinin boyutu bildirir): Kullanılan değişken dizisinin boyutunu yani kaç değişkenden oluştuğunu tanımlar.

DLY (zaman geciktirme): Bekleme zamanını ayarlar yada çıkış sinyalinin darbe uzunluğunu belirtir.

CALLP (program çağır): Kontrolörün hafızasından adını kullanarak bir program çağırmak için kullanılır.

END (son): Programın çalışmasını sonlandırır.

IF-THEN-ELSE : Belirtilen şartın sorgulanması neticesinde sonucun doğru ya da yanlış olmasına göre dallanma yapar.

SELECT-CASE: Sorgulanan birçok şarta göre farklı blokları çalıştırır. Aynı değişken birden çok durum için sorgulanabilir ve ilişki yazılabilir. Örneğin A değeri 0' dan büyükse 1'e eşitse 2'den küçükse şeklinde girilen her şart için farklı bir blok çalıştırılabilir.

FOR-NEXT: Deyimler arasında kalan program parçası sayacın belirttiği sayı kadar tekrarlanır.

WHILE-WEND: Şart sağlanana kadar döngü deyimleri arasında kalan işlemler tekrarlanır.

WAIT: Programı sorgulanan değişken, değerine ulaşmaya kadar bekletilir.

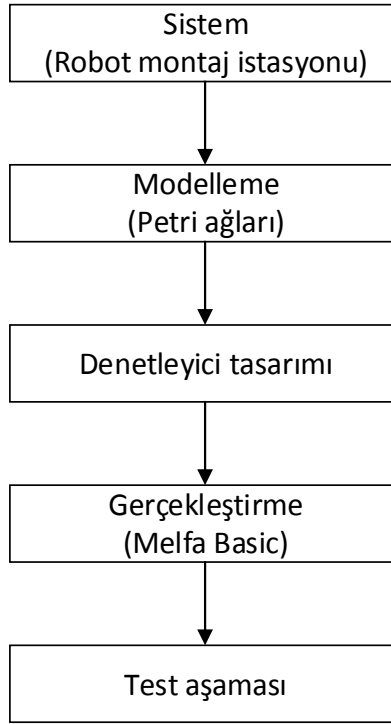
GOSUB: Satır numarası ya da adı verilen alt programı çağırır.

Programda kullanılan *RETURN* komutu alt programın bitiminde kullanıldığında alt programın başladığı satırın alt satırına gönderir. Program oradan devam eder.

GOTO: Herhangi bir şart belirtmeden istenilen satır numarasına ya da etiketi (ismi) belirtilmiş bir satıra atlamayı sağlar.

4. BULGULAR VE TARTIŞMA

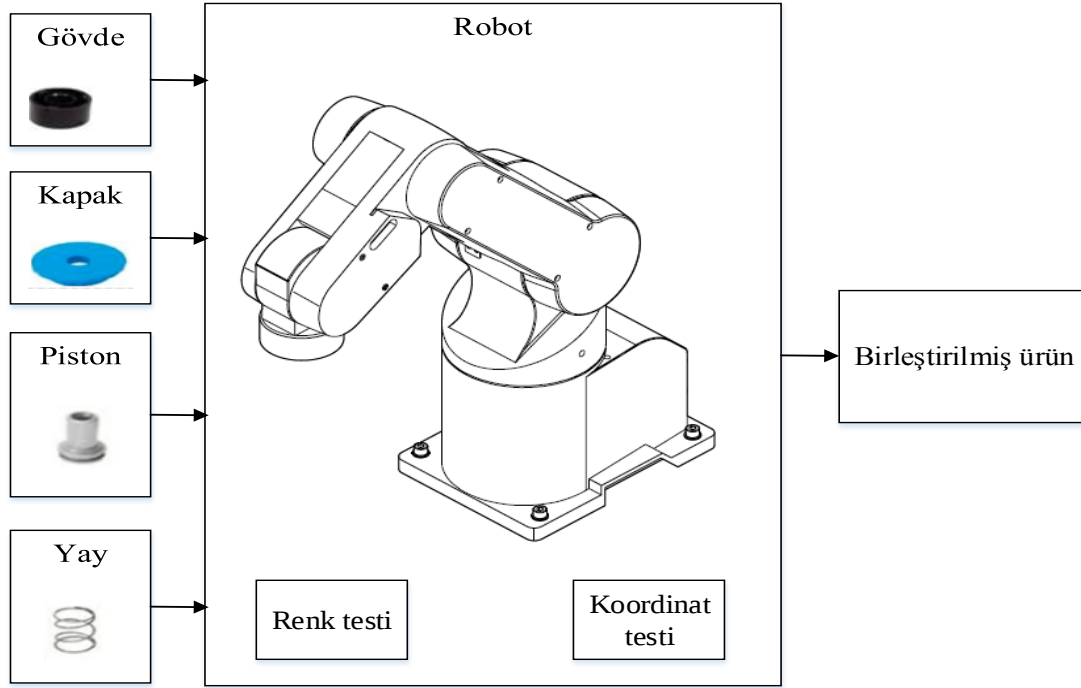
Bu tez çalışmasında izlenen çalışma planı blok diyagram olarak Şekil 4.1’de verilmiştir. Bu tez de Robotik Montaj Sistemlerinin Petri Ağı Temelli Kontrolü yapılacaktır. İlk olarak Petri ağı modeli oluşturularak sonrasında sistem için gözetici hesaplanacak son olarak Melfa Basic yazılım programı ile denetleyicinin gerçekleştirilmesi yapılmaktadır.



Şekil 4.1. Tez kapsamında gerçekleştirilecek çalışmalar

4.1. Sistem Petri Ağı Modeli

FESTO tarafından üretilmiş olan deneysel Robotik Montaj istasyonunun çalışması tezin üçüncü bölümünde anlatılmıştır. Sistemin genel şeması Şekil 4.2’de görülmektedir.



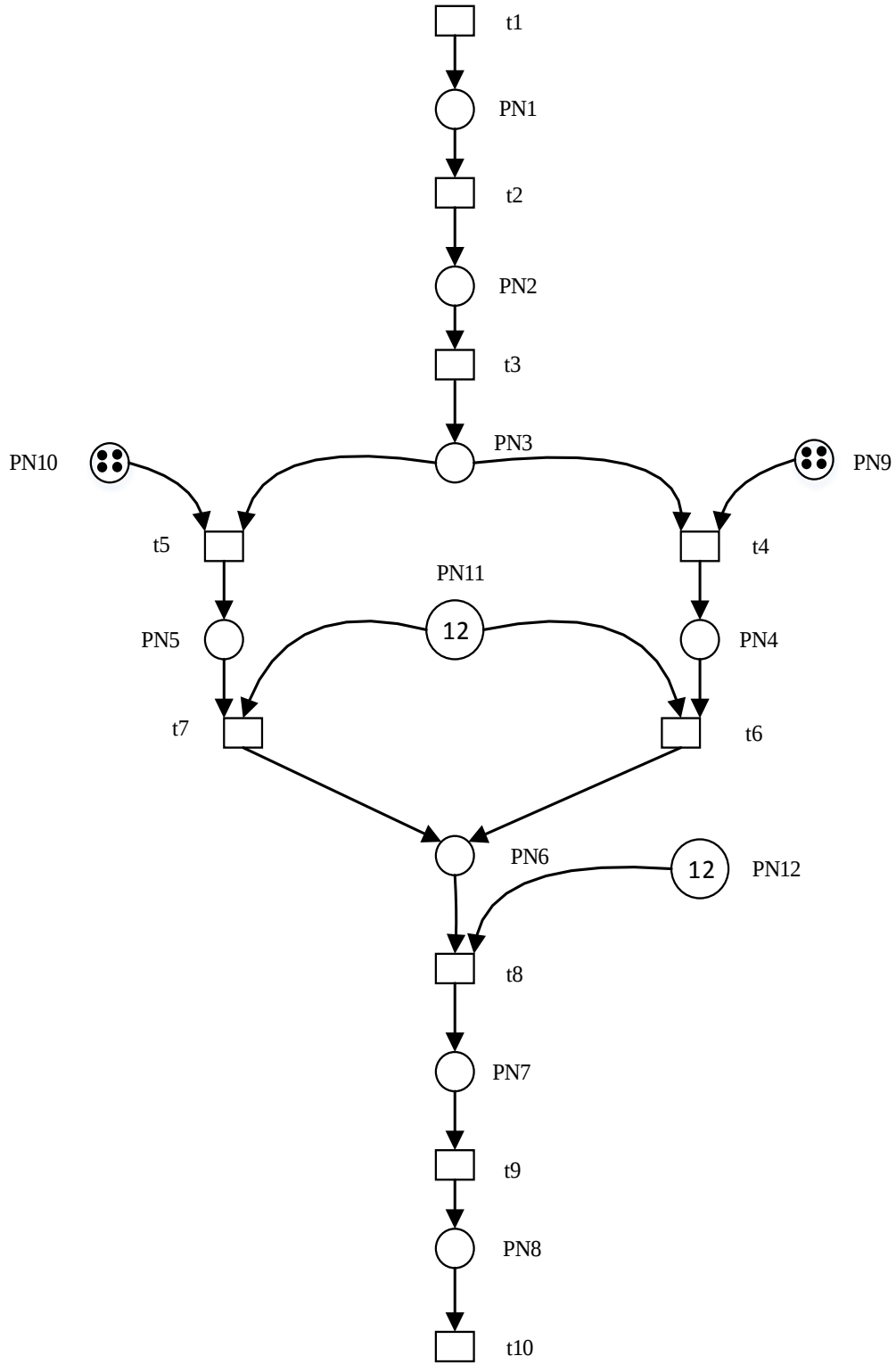
Şekil 4.2. Robot veri istasyonu genel şeması

Robotik montaj istasyonunun çalışması dikkate alınarak elde edilen kontrol edilmemiş Petri ağı modeli Şekil 4.3'te sunulmuştur. Petri ağı modeli oluşturulurken, robotun yerine getirdiği işlemler gruplanarak TASK1, TASK2, ..., TASK7 olarak adlandırılmıştır. Bu robotik görevlerin ilgili PA mevkisinde jeton bulunduğu durumda gerçekleştirileceği, bulunmadığı durumda ise çalıştırılmayacağı planlanmıştır. Robotik görevlerin açıklaması Çizelge 4.1'de sunulmuştur. Petri ağı modelinde geçiş ve mevki açıklamaları sırasıyla Çizelge 4.2 ve Çizelge 4.3'te verilmiştir.

Sistemin Petri ağı modelinde PN1, PN2, ..., PN12 olmak üzere toplam 12 mevki ve $t_1, t_2, \dots, t_{10}$ olmak üzere toplam 10 geçiş bulunmaktadır. Başlangıçta PN9 ve PN10 mevkilerinde 4'er PN11 ve PN12 mevkilerinde ise 12'şer adet jeton mevcuttur. PN9 ve PN10 mevkileri sırasıyla siyah ve metal piston paletlerini temsil etmektedir. Deneysel montaj istasyonunda 4'er adet piston için yer olduğundan modelde başlangıçta 4'er adet jeton mevcuttur. Benzer olarak PN11 ve PN12 mevkilerinde yay ve kapak magazinlerini temsil etmektedir ve başlangıçta 12'şer adet jeton yüklenmiştir.

Şekil 4.3'te verilen robotik montaj istasyonu Petri ağı modelinin çalışması şu şekildedir. t_1 geçişi silindirin giriş rayına geldiği bilgisini veren PART_AV sensörü ile tetiklenmektedir. Tetiklenme gerçekleştiğinde PN1 mevkisine bir adet jeton aktarılmaktadır. Bu TASK1 görevinin yerine getirileceğini belirtmektedir. Yani silindir robot tarafından alınarak test

alanına bırakılır. t2 geçişi TASK1 işleminin bitmesi şartı ile tetiklenir. PN1 mevkisindeki jeton PN2 mevkisine aktarıldığında TASK2 görevi yerine getirilir. Yani robot silindire renk ve koordinat testini yaptıktan sonra silindiri alarak birleştirme alanına yerleştirir. t3 geçişi TASK2 işleminin bitmesiyle tetiklenir. PN2 mevkisindeki jeton PN3 mevkisine geçer ve bu mevki TASK1 ve TASK2 işlemlerinin bittiğini, silindirin birleştirme alanında olduğunu göstermektedir. TASK2 işlemi içerisinde parçanın siyah olup olmadığı bilgisi elde edilmiştir robot üzerindeki renk algılama sensörü ile parça siyah değil ise ve PN9 mevkisinde en az bir adet jeton mevcut ise t4 geçişi tetiklenir. t4 geçişinin tetiklenmesiyle PN3 ve PN9 mevkilerinden jeton eksiltilerek PN4 mevkisine bir adet jeton eklenir. Jeton PN4 mevkisinde iken TASK3 görevi yerine getirilir. Yani robot piston paletine giderek bir adet siyah pistonu alıp birleştirme alanına getirir. TASK3 işleminin bitmesi şartına ve PN11 mevkisinde en az bir jeton olması şartına bağlı olarak t6 geçişi tetiklenir ve jeton PN6 mevkisine akmaktadır. Silindir siyah ise ve PN10 mevkisinde en az bir adet jeton mevcut ise t5 geçişi tetiklenir. t5 geçişinin tetiklenmesiyle PN3 ve PN10 mevkilerinden jeton eksiltilerek PN5 mevkisine bir adet jeton eklenir. Jeton PN5 mevkisinde iken TASK4 görevi yerine getirilir. Yani robot piston paletine giderek bir adet metal piston alıp birleştirme alanına getirir. TASK4 görevinin bitmesi ve PN11 mevkisinde en az bir jeton olması şartına bağlı olarak t7 geçişi tetiklenir ve jeton PN6 mevkisine akmaktadır. Jeton PN6 mevkisinde iken TASK5 işlemi gerçekleştirilir. Yani robot yay magazinine giderek bir adet yay alıp birleştirme alanına getirmektedir. t8 geçişi TASK5 işleminin bitmesi ve PN12 mevkisinde en az bir adet jeton olması şartına bağlı olarak tetiklenir. PN6 ve PN12 mevkilerinden jeton eksiltilerek PN7 mevkisine bir adet jeton eklenir. Jeton PN7 mevkisinde iken TASK6 görevi yerine getirilir. Yani robot kapak magazinine giderek bir adet kapak alıp koordinat testini yaptıktan sonra kapağı silindir üzerine monte etmektedir. t9 geçişi TASK6 görevinin yerine getirilmesi şartına bağlı olarak tetiklenir. PN7 mevkisinden bir adet jeton eksiltilerek PN8 mevkisine eklenir. Jeton PN8 mevkisinde iken TASK7 görevi yerine getirilir. Yani robot birleştirilmiş ürünü alıp birleştirilmiş ürün rayına bırakmaktadır. t10 geçişi TASK7 görevinin yerine getirilmesiyle tetiklenir ve montajda parça kalmadığını ifade etmektedir.



Şekil 4.3. Robotik montaj istasyonu Petri ağı modeli

Çizelge 4.1. Robotik görevler ve açıklamaları

Robotik görev adı	Açıklama
TASK1	Silindir giriş rayından alınarak renk test alanına bırakılır
TASK2	Silindire renk ve koordinat testi yaptırılarak birleştirme alanına bırakılır
TASK3	Siyah piston, piston paletinden alınarak birleştirme alanına bırakılır
TASK4	Metal piston, piston paletinden alınarak birleştirme alanına bırakılır
TASK5	Yay, yay magazininden alınarak birleştirme alanına bırakılır
TASK 6	Kapak, kapak magazininden alınarak koordinat testi yapıldıktan sonra birleştirme alanına bırakılır
TASK 7	Montajı yapılan ürün birleştirilmiş ürün rayına bırakılır

Çizelge 4.2. Petri ağı modelindeki mevkilerin açıklamaları

Mevkiler	Açıklama
PN1	TASK1
PN2	TASK2
PN3	Silindir birleştirme alanında montaja hazır
PN4	TASK3
PN5	TASK4
PN6	TASK5
PN7	TASK6
PN8	TASK7
PN9	Siyah piston paleti (en fazla 4 adet)
PN10	Metal piston paleti (en fazla 4 adet)
PN11	Yay magazini (en fazla 12 adet)
PN12	Kapak magazini (en fazla 12 adet)

Çizelge 4.3. Petri ağı modelindeki geçişlerin açıklamaları

Geçişler	Harici tetikleme koşulu (sensör bilgisi)	Açıklama
t1	Part_Av	Silindir giriş rayına geldi
t2		TASK1 tamamlandı
t3		TASK2 tamamlandı
t4	CYLTTYPE=0	Silindir metal ya da kırmızı
t5	CYLTTYPE=1	Silindir siyah
t6		TASK3 tamamlandı
t7		TASK4 tamamlandı
t8		TASK5 tamamlandı
t9		TASK6 tamamlandı
t10		TASK7 tamamlandı

4.2. Sistem Gözeticisinin Hesaplanması

Robotik montaj istasyonu için gözeticiler tezin üçüncü bölümünde anlatılmış olan mevki değişmezleri ile gözetici sentezlenmesi metodu kullanılarak hesaplanmıştır. Sistem kontrolü için bazı spesifikasyonlar belirlenerek bu spesifikasyonlar için kontrol mevkileri hesaplanmıştır. Belirlenen kontrol spesifikasyonları için kontrol mevkisi hesapları aşağıda açıklanmaktadır.

Spec 1: Sistemde sadece bir adet robot olduğundan dolayı herhangi bir anda robotik görevlerden sadece bir tanesi yerine getirilebilir.

Spec 2: Sistemde aynı anda sadece tek bir parça montajı yapılabilir.

Spec 3: Sistemde aynı anda sadece tek bir silindirin montajı yapılabileceğinden bir işlemde metal veya siyah pistondan sadece bir tanesi kullanılabilir.

Spec 4: Sistemde siyah piston veya metal piston kalmamış ise silindir montajı tamamlanamayacağından yeni silindir işleme alınmamalıdır.

Spec 5: Sistemde yay kalmamış ise silindir montajı tamamlanamayacağından yeni silindir işleme alınmamalıdır.

Spec 6: Sistemde kapak kalmamış ise silindir montajı tamamlanamayacağından yeni silindir işleme alınmamalıdır.

Şekil 4.3'te verilen Petri ağı modeline ilişkin D_p çakışıklık (incidence) matrisi ve μ_{p0} başlangıç işaretlemesi vektörü model üzerinden aşağıdaki şekilde elde edilmiştir.

$$D_p = \begin{bmatrix} +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

$$\mu_{p0} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \\ \mu_8 \\ \mu_9 \\ \mu_{10} \\ \mu_{11} \\ \mu_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 4 \\ 4 \\ 12 \\ 12 \end{bmatrix}$$

Belirlediğimiz spesifikasyonlara ilişkin kontrolör hesapları aşağıda görülmektedir.

Spec 1: Sistemde sadece bir adet robot olduğundan dolayı herhangi bir anda robotik görevlerden sadece bir tanesinin yerine getirilebilmesi için gerekli şart mevki değişmezi olarak aşağıdaki gibi ifade edilebilir;

$$\mu_1 + \mu_2 + \mu_4 + \mu_5 + \mu_6 + \mu_7 + \mu_8 \leq 1$$

Yukarıdaki kısıttan da anlaşılabileceği üzere robotik görevler PA modelinde PN1, PN2, PN4, PN5, PN6, PN7 ve PN8 mevkileri ile ilişkilidir. Dolayısıyla bu mevkilerdeki jetonların toplamının 1'den küçük veya eşit olduğu durumda ilgili mevkileri sadece bir tanesinde jeton bulunabileceği anlaşılır. Bu da ilgili kontrol spesifikasyonunun doğru modellendiğini göstermektedir. Bu şartı sağlayan C_1 kontrol mevkisi L_1 sistem sınırlayıcısı, μ_{c1} işaret ataması ve D_{c1} çakışıklık (incidence) matrisi hesaplamaları aşağıda görülmektedir.

$$\mu_1 + \mu_2 + \mu_4 + \mu_5 + \mu_6 + \mu_7 + \mu_8 \leq 1$$

$$b \leq 1$$

$$L_1 = [1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]$$

$$\mu_{c1} = b - L_1 \cdot \mu_{p1}$$

$$\mu_{c1} = 1 - [1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0] \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 4 \\ 4 \\ 12 \\ 12 \end{bmatrix}$$

$$\mu_{c1} = 1 - 0$$

$$\mu_{c1} = 1$$

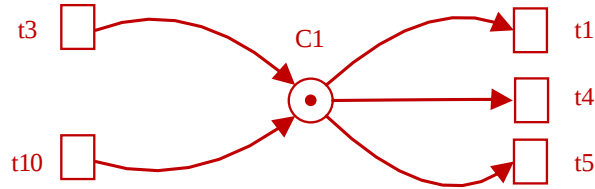
$$D_{c1} = - L_1 \cdot D_p$$

$$D_{c_1} = -[1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0].$$

$$\begin{bmatrix} +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

$$D_{c_1} = [-1 \ 0 \ +1 \ -1 \ -1 \ 0 \ 0 \ 0 \ 0 \ +1]$$

D_{c_1} vektöründen de görülebileceği üzere kontrol spesifikasyonunu sağlayacak C_1 kontrol mevkisinin giriş geçişleri t3 ve t10, çıkış geçişleri ise t1, t4 ve t5'dir. Başlangıçta ise bir adet jeton içermektedir. Hesaplanan C_1 kontrol mevkisinin PA modeli Şekil 4.4'te görülmektedir.



Şekil 4.4. C_1 kontrol mevkisi ve ilgili geçişleri

Spec 2: Sistemde aynı anda sadece tek bir parça montajının yapılabilmesi için gerekli şart şu şekildedir;

$$\mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_7 + \mu_8 \leq 1$$

Yukarıdaki kısıttan da anlaşılabileceği üzere robotik görevler PA modelinde PN1, PN2, PN3, PN4, PN5, PN6, PN7 ve PN8 mevkileri ile ilişkilidir. Dolayısıyla bu mevkilerdeki jetonların toplamının 1'den küçük veya eşit olduğu durumda ilgili mevkilerin sadece bir tanesinde jeton bulunabileceği anlaşılır. Bu da ilgili kontrol spesifikasyonunun doğru modellendiğini

göstermektedir. Bu şartı sağlayan C_2 kontrol mevkisinin L_2 sistem sınırlayıcısı, μ_{c2} işaret ataması ve D_{c2} çakışıklık (incidence) matrisi hesaplamaları aşağıda görülmektedir.

$$\mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_7 + \mu_8 \leq 1$$

$$b \leq 1$$

$$L_2 = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]$$

$$\mu_{c2} = b - L_2 \cdot \mu_{p2}$$

$$\mu_{c2} = 1 - [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0] \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 4 \\ 4 \\ 12 \\ 12 \end{bmatrix}$$

$$\mu_{c2} = 1 - 0$$

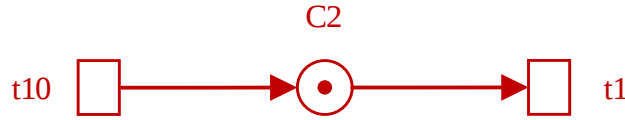
$$\mu_{c2} = 1$$

$$D_{c2} = -L_2 \cdot D_p$$

$$D_{c_2} = -[1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0] \cdot \begin{bmatrix} +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

$$D_{c_2} = [-1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ +1]$$

D_{c_2} vektöründen de görülebileceği üzere kontrol spesifikasyonunu sağlayan C2 kontrol mevkisinin giriş geçişi t10 ve çıkış geçişi t1'dir. Başlangıçta ise bir adet jeton içermektedir. Hesaplanan C2 kontrol mevkisinin PA modeli Şekil 4.5'de görülmektedir.



Şekil 4.5. C2 kontrol mevkisi ve ilgili geçişleri

Spec 3: Sistemde aynı anda sadece tek bir silindirin montajı yapılabileceğinden bir işlemden metal veya siyah pistondan sadece bir tanesinin kullanılabilmesi için gerekli şart şu şekildedir;

$$\mu_4 + \mu_5 \leq 1$$

Yukarıdaki kısıttan da anlaşılabileceği üzere robotik görevler PA modelinde PN4 ve PN5 mevkileri ile ilişkilidir. Dolayısıyla bu mevkilerdeki jetonların toplamının 1'den küçük veya eşit olduğu durumlarda ilgili mevkilerin sadece bir tanesinde jeton bulunabileceği anlaşılır. Bu da ilgili kontrol spesifikasyonunun doğru modellendiğini göstermektedir. Bu şartı sağlayan C_3 kontrol mevkisinin L_3 sistem sınırlayıcısı, μ_{c_3} işaret ataması ve D_{c_3} çakışıklık (incidence) matrisi hesaplamaları aşağıda görülmektedir.

$$\mu_4 + \mu_5 \leq 1$$

$$b \leq 1$$

$$L_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mu_{c3} = b - L_3 \cdot \mu_{p3}$$

$$\mu_{c3} = 1 - \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 4 \\ 4 \\ 12 \\ 12 \end{bmatrix}.$$

$$\mu_{c3} = 1 - 0$$

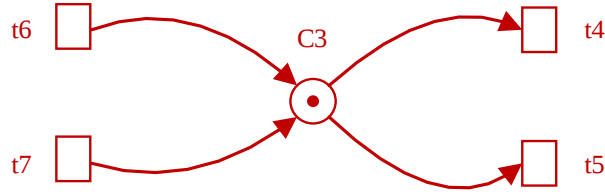
$$\mu_{c3} = 1$$

$$D_{c_3} = - L_3 \cdot D_p$$

$$D_{c_3} = - \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & +1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}.$$

$$D_{c_3} = [0 \ 0 \ 0 \ -1 \ -1 \ +1 \ +1 \ 0 \ 0 \ 0]$$

D_{c_3} vektöründen de görülebileceği üzere kontrol spesifikasyonunu sağlayan C3 kontrol mevkisinin giriş geçişleri t6 ve t7, çıkış geçişleri ise t4 ve t5'dir. Başlangıçta ise bir adet jeton içermektedir. Hesaplanan C3 kontrol mevkisinin PA modeli Şekil 4.6'da görülmektedir.



Şekil 4.6. C3 kontrol mevkisi ve ilgili geçişleri

Spec 4: Sistemde siyah piston veya metal piston kalmamış ise silindir montajı tamamlanamayacağından yeni silindirin işleme alınmaması için gerekli şart şu şekildedir;

$$\mu_9 + \mu_{10} - \mu_1 \geq 0$$

Yukarıdaki kısıttan da anlaşılabileceği üzere robotik görevler PA modelinde PN1, PN9 ve PN10 mevkileri ile ilişkilidir. Dolayısıyla PN10 ve PN9 mevkilerindeki jetonların toplamından PN1 mevkisindeki jeton sayısı çıkarıldığında kalan jeton miktarının sıfıra eşit yada büyük olduğu durumlarda PN9 veya PN10'da ve PN1'de jeton bulunabileceği anlaşılır. Bu da ilgili kontrol spesifikasyonunun doğru modellendiğini göstermektedir. Bu şartı

sağlayan C_4 kontrol mevkisinin L_4 sistem sınırlayıcısı, μ_{c4} işaret ataması ve D_{c4} çakışıklık (incidence) matrisi hesaplamaları aşağıda görülmektedir.

$$\mu_1 - \mu_9 - \mu_{10} \leq 0$$

$$b \leq 0$$

$$L_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \end{bmatrix}$$

$$\mu_{c4} = b - L_4 \cdot \mu_{p4}$$

$$\mu_{c4} = 0 - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 4 \\ 4 \\ 12 \\ 12 \end{bmatrix}$$

$$\mu_{c4} = 0 - (-12)$$

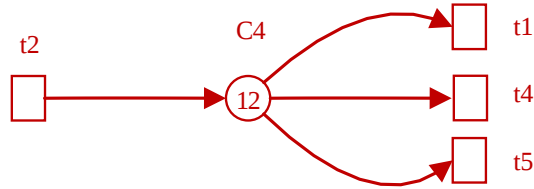
$$\mu_{c4} = 12$$

$$D_{c4} = - L_4 \cdot D_p$$

$$D_{c_4} = - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & +1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

$$D_{c_4} = [-1 \quad +1 \quad 0 \quad -1 \quad -1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

D_{c_4} vektöründen de görülebileceği üzere kontrol spesifikasyonunu sağlayan C4 kontrol mevkisinin giriş geçişi t2, çıkış geçişleri ise t1, t4 ve t5'dir. Başlangıçta ise 12 adet jeton içermektedir. Hesaplanan C4 kontrol mevkisinin PA modeli Şekil 4.7'de görülmektedir.



Şekil 4.7. C4 kontrol mevkisi ve ilgili geçişleri

Spec 5: Sistemde yay kalmamış ise silindir montajı tamamlanamayacağından yeni silindirin işleme alınmaması için gerekli şart şu şekildedir;

$$\mu_{11} - \mu_1 \geq 0$$

Yukarıdaki kısıttan da anlaşılabileceği üzere robotik görevler PA modelinde PN1 ve PN11 mevkileri ile ilişkilidir. Dolayısıyla PN11 mevkisindeki jeton sayısından PN1 mevkisindeki jeton sayısı çıkarıldığında kalan jeton miktarının sıfıra eşit yada büyük olduğu durumlarda PN11 ve PN1 mevkilerinde en az bir jeton bulunabileceği anlaşılır. Bu da ilgili kontrol

spesifikasyonunun doğru modellendiğini göstermektedir. Bu şartı sağlayan C_5 kontrol mevkisinin L_5 sistem sınırlayıcısı, μ_{c5} işaret ataması ve D_{c5} çakışıklık (incidence) matrisi hesaplamaları aşağıda görülmektedir.

$$\mu_1 - \mu_{11} \leq 0$$

$$b \leq 0$$

$$L_5 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1 \ 0]$$

$$\mu_{c5} = b - L_5 \cdot \mu_{p5}$$

$$\mu_{c5} = 0 - [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1 \ 0] \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 4 \\ 4 \\ 12 \\ 12 \end{bmatrix}$$

$$\mu_{c5} = 0 - (-12)$$

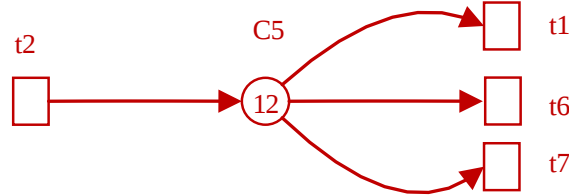
$$\mu_{c5} = 12$$

$$D_{c5} = -L_5 \cdot D_p$$

$$D_{c_5} = - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

$$D_{c_5} = [-1 \quad +1 \quad 0 \quad 0 \quad 0 \quad -1 \quad -1 \quad 0 \quad 0 \quad 0]$$

D_{c_5} vektöründen de görülebileceği üzere kontrol spesifikasyonunu sağlayan C5 kontrol mevkisinin giriş geçişi t2, çıkış geçişleri ise t1, t6 ve t7'dir. Başlangıçta ise 12 adet jeton içermektedir. Hesaplanan C5 kontrol mevkisinin PA modeli Şekil 4.8'de görülmektedir.



Şekil 4.8. C5 kontrol mevkisi ve ilgili geçişleri

Spec 6: Sistemde kapak kalmamış ise silindir montajı tamamlanamayacağından yeni silindirin işleme alınmaması için gerekli şart şu şekildedir;

$$\mu_{12} - \mu_1 \geq 0$$

Yukarıdaki kısıttan da anlaşılabileceği üzere robotik görevler PA modelinde PN1 ve PN12 mevkileri ile ilişkilidir. Dolayısıyla PN12 mevkisindeki jeton sayısından PN1 mevkisindeki jeton sayısı çıkarıldığında kalan jeton miktarının sıfıra eşit yada büyük olduğu durumlarda PN12 ve PN1 mevkilerinde en az bir jeton bulunabileceği anlaşılır. Bu da ilgili kontrol spesifikasyonunun doğru modellendiğini göstermektedir. Bu şartı sağlayan C_6 kontrol

mevkisini L_6 sistem sınırlayıcısı, μ_{c6} işaret ataması ve D_{c6} çakışıklık (incidence) matrisi hesaplamaları aşağıda görülmektedir.

$$\mu_1 - \mu_{12} \leq 0$$

$$b \leq 0$$

$$L_6 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1]$$

$$\mu_{c6} = b - L_6 \cdot \mu_{p6}$$

$$\mu_{c6} = 0 - [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1] \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 4 \\ 4 \\ 12 \\ 12 \end{bmatrix}$$

$$\mu_{c6} = 0 - (-12)$$

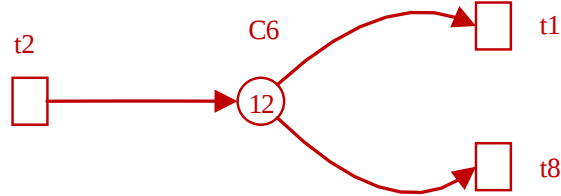
$$\mu_{c6} = 12$$

$$D_{c6} = - L_6 \cdot D_p$$

$$D_{c_6} = - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & +1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

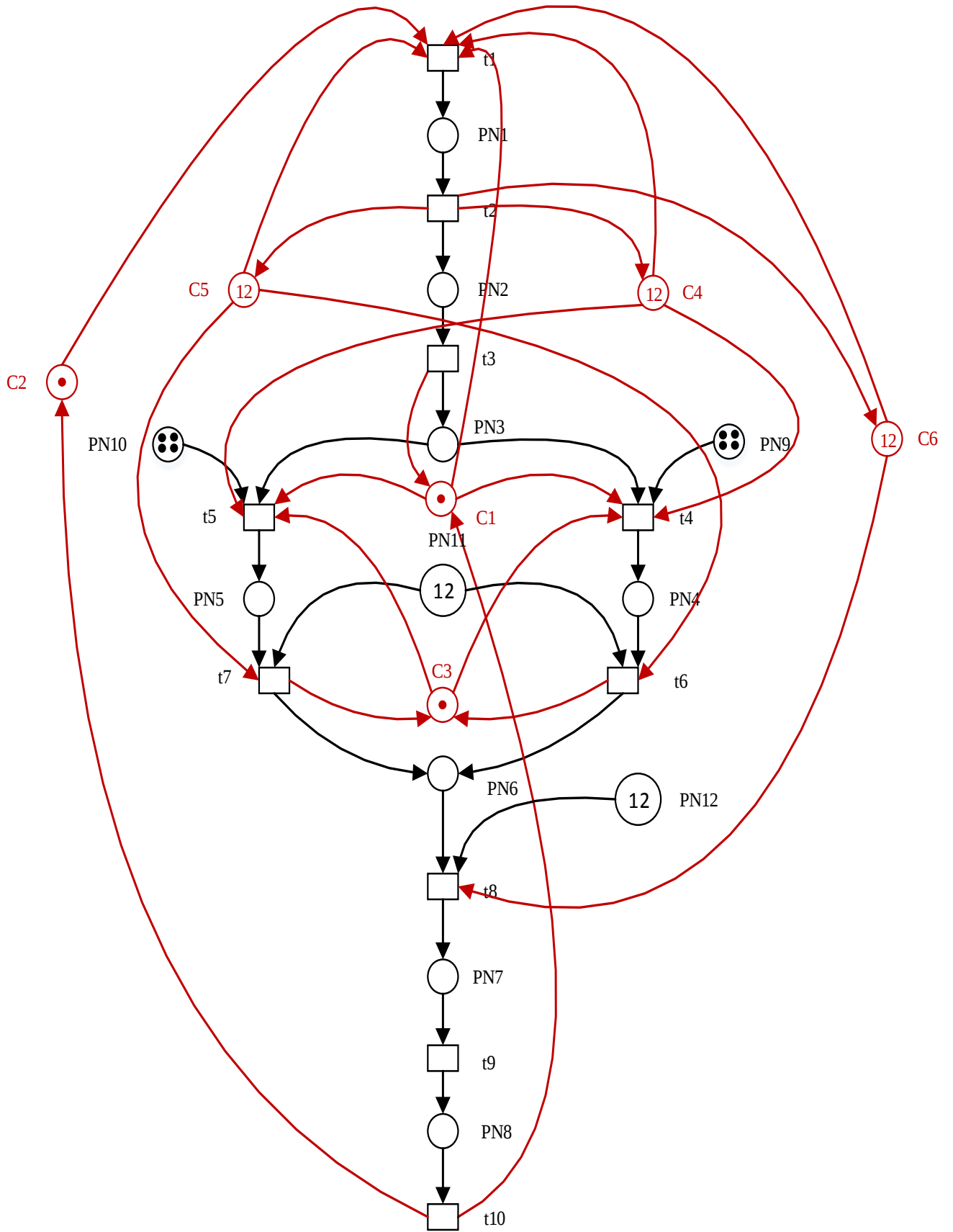
$$D_{c_6} = [-1 \quad +1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad -1 \quad 0 \quad 0]$$

D_{c_6} vektöründen de görülebileceği üzere kontrol spesifikasyonunu sağlayan C5 kontrol mevkisinin giriş geçişi t2, çıkış geçişleri ise t1 ve t8'dir. Başlangıçta ise 12 adet jeton içermektedir. Hesaplanan C6 kontrol mevkisinin PA modeli Şekil 4.9'da görülmektedir.



Şekil 4.9. C4 kontrol mevkisi ve ilgili geçişler

Tüm kontrol spesifikasyonları için elde edilen kontrol mevkileri Şekil 4.3'te görülen PA modeline eklenmesi ile elde edilen kontrol edilmiş PA modeli Şekil 4.10'da görülmektedir.



Şekil 4.10. Robotik montaj istasyonunun kontrol edilmiş PA modeli

4.3. Petri Ağı Modelinin Melfa Basic Programlama Dili ile Gerçekleştirilmesi

Mitsubishi RV- 2SDB Robot denetleyicisi için kontrol kodu Melfa BASIC programlama dili ile sağlanmaktadır. Bu dilin detayları önceki bölümlerde açıklanmıştır. Melfa Basic dili ardışıl programlamaya olanak sağlayan bir programlama yapısına sahiptir. Petri ağlarının bu dil ile gerçekleştirilmesinde döngüsel bir kod içerisinde belirli hafıza alanlarına kayıtlı değişkenlerdeki değerlerin atırılıp azaltılması amaçlanmaktadır. Her bir mevki için tam sayı (integer) tipinde bir hafıza değişkeni tanımlanarak ilgili geçişin tetiklenme koşulunun sağlanmasının ardından PA doğasına uygun davranışsal özellik mevkileri temsil eden değişkenler üzerinde gerçekleştirilmektedir. Gerçekleştirme işleminin daha anlaşılır olması için Şekil 4.11’de görülen basit PA modeli örnek olarak dikkate alınmıştır.

Bu modelde bir adet mevki ve iki adet geçiş mevcuttur. t1 geçişinin tetiklenmesiyle p1 mevkisine jeton aktarılmakta t2 geçişinin tetiklenmesiyle ise jeton mevkiden çıkarılmaktadır. p1 mevkisinin bir robotik görevi (TASK) ifade ettiği ve içerisinde jeton bulunduğu görevin yerine getirildiği, jeton olmadığı ise görevin yerine getirilmediği varsayılın. t1 geçişinin harici bir sensör bilgisinde tetikleneceği kabul edilsin. t2 geçişinin ise TASK1’in tamamlanmasından sonra tetiklenebildiği farz edilsin.

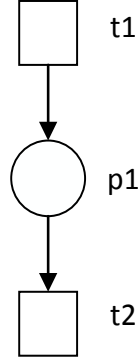
PA modelleri Melfa Basic dili ile gerçekleştirilirken Şekil 4.13’de görülen program yapısı kullanılmaktadır. Buna göre ilk olarak mevki vb. için değişken tanımlamaları yapılmalıdır. Daha sonra PA için başlangıç jeton sayısı atamaları gerçekleştirilir. Kodun buraya kadar olan kısmı döngü içerisinde olmadığından sadece bir defa çalıştırılacaktır. İlgili işlemler ardından kod içerisindeki dinamik yapının elde edildiği döngü kurulmuştur. Döngü içerisinde sırasıyla, Aksiyon atamaları, PA geçişleri ve görev tamamlandı bilgilerinin resetlenmesi işlemleri yerine getirilmektedir.

Aksiyon atamaları kısmında PA modelinde bulunan mevkilerle tanımlı görevlerin ataması yapılmaktadır. Yani hangi mevkiden hangi görev yetkilendirilecekse bu işlemler bu kısımda tanımlanmalıdır.

PA geçişleri kısmında, PA modelinin dinamik yapısı kod olarak gerçekleştirilir. Yani her geçişin tetiklenmesiyle PA modelindeki jeton hareketleri kod olarak değişkenler üzerinde taklit edilir.

Görev tamamlandı bilgileri ilgili TASK alt programına gidildiğinde resetlenir. Bu işlem ile döngü içerisinde TASK alt programına sadece bir defa gitmesi garantilenir. Alt programdan çıkılırken görev tamamlandı bilgisi set edilir. İlgili PA geçişi tetiklendikten sonra kodun son

bölümünde ise tüm görev tamamlandı bilgileri resetlenmektedir. Şekil 4.11’de görülen basit PA modeli için oluşturulan Melfa Basic kodu aşağıda verilmiştir.



Şekil 4.11. Örnek Petri ağı modeli

```
'----- mevki tanımlamaları-----'

Def Inte P1

'----- geçiş tanımlamaları-----'

Def Inte sensör1
Def Inte T1_FNS

'----- başlangıç değer atamaları-----'

PN1 = 0
T1_FNS = 0

GoTo *Başla
END

'----- program kısmı -----'

Başla

'----- aksiyon atamaları -----'

If sensör1 = 1 Then
PN1 = 1
ENDIF

If (PN1 = 1 AND T1_FNS = 1 ) Then
PN1 = 0
ENDIF

'----- Alt Program Atamaları -----'

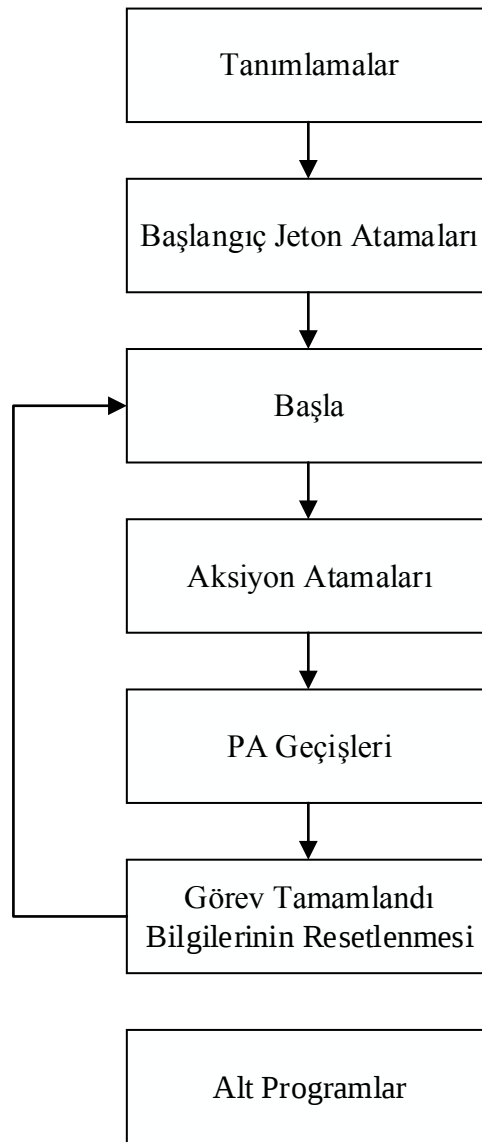
*TASK1
If PN1 = 1 Then GoSub *TASK1

*TASK1
```

```

T1_FNS = 0
-----
-----
-----
T1_FNS = 1
Return

```



Şekil 4.12. PA modelinin Melfa Basic ile gerçekleştirilirken kullanılacak program yapısı

4.4. Robotik Montaj İstasyonu PA Modelinin Gerçekleştirilmesi

Şekil 4.10' da verilmiş olan Robotik montaj istasyonunun kontrol edilmiş Petri ağı modelinin Melfa Basic dili kullanılarak gerçekleştirilmesi yapılmıştır. Bu gerçekleştirme esnasında ilk olarak ilgili pozisyon atamaları ve bu pozisyonlara ait vektör atamaları yapılmıştır. Sonrasında, robotun harici adres tanımlamaları, integer tanımlamaları, palet tanımlamaları, mevki tanımlamaları, denetleyici tanımlamaları, geçiş tanımlamaları yapılmıştır. Daha sonra, ilgili PA modeline ilişkin mevkilerde ve gözeticilerde bulunan başlangıç jeton atamaları yapılmıştır ve robotun başlama pozisyonu olan P99 pozisyonuna yönlendirilmesi yapılmıştır ve bu işlemlerin ardından ilgili mevkilerde hangi görevin yerine getirileceğine ilişkin görev çağırımları yapılarak geçiş şartları ve başlangıç geçiş değer atamaları yapılarak alt programlara geçiş sağlanmıştır.

Bu tez de kullanıcı tarafından oluşturulan on adet pozisyon vardır. Bu pozisyonlar aşağıda sunulmuştur.

```
DEF POS P99=(+196.36,-14.48,+369.21,-179.97,-0.03,+179.17,+0.00,+0.00)(7,0)
DEF POS P1=(+231.20,+64.91,+180.79,+179.99,+0.00,-90.00,+0.00,+0.00)(7,0)
DEF POS P2=(+216.66,+211.79,+174.97,-180.00,+0.00,-180.00,+0.00,+0.00)(7,0)
DEF POS P3=(+184.87,+232.07,+154.38,-180.00,+0.00,+180.00,+0.00,+0.00)(7,0)
DEF POS P4=(+204.23,+262.81,+155.92,+179.97,+0.29,-133.95,+0.00,+0.00)(7,0)
DEF POS P5=(-141.71,+276.88,+156.93,-180.00,+0.00,+180.00,+0.00,+0.00)(7,0)
DEF POS P6=(+135.38,+353.45,+328.73,-180.00,+0.00,+90.00,+0.00,+0.00)(7,0)
DEF POS P7=(-209.76,+262.95,+178.21,+180.00,+0.00,+52.24,+0.00,+0.00)(7,0)
DEF POS P8=(-78.54,+203.15,+354.17,-179.98,-0.01,-90.21,+0.00,+0.00)(7,0)
DEF POS P9=(+66.25,+297.28,+157.78,-179.99,+0.01,+140.91,+0.00,+0.00)(7,0)
```

Kontrol edilmiş PA modeline ilişkin gerçekleştirilen sistemin başarıyla kontrol edildiği gözlemlenmiştir. Sisteme ait Melfa Basic kodu aşağıda verilmiştir.

'----- pozisyon atamaları-----

```
Def Pos VECT30      'silindir kapağını döndürme açısı
Def Pos VECY50      'y-ekseninde 50 mm git
Def Pos VECZ2       'z-ekseninde 2.5 mm git
Def Pos VECZ5       'z-ekseninde 5 mm git
Def Pos VECZ30      'z-ekseninde 30 mm git
Def Pos VECZ50      'z-ekseninde 50 mm git
Def Pos VECZ100     'z-ekseninde 100 mm git
Def Pos VECZ200     'z-ekseninde 200 mm git
Def Pos VECZINT     'z- ekseninde iç açı ölçüsü
Def Pos VECGA       'silindir pozisyonun doğrulama
Def Pos AUXPOS      'yedek pozisyon
Def Pos VECMINC     'iç oryantasyon ölçüsü
Def Pos VECCACYL    'oryantasyon ölçüsünü doğrulama açısı
Def Pos VECCACOV    'oryantasyon ölçüsünü doğrulama açısı
Def Pos VECEP1      'palet 1 ' in son noktası
Def Pos VECEP2      'palet 2 ' nin son noktası
Def Pos VECCUP      'tutucuya öğretme pozisyonu
Def Pos VECCMPZ     'öğretme pozisyonu doğrulama
Def Pos VECCOZ      'silindir oryantasyon doğrulama pozisyonu
Def Pos VECCOD      'kapak oryantasyon öğretme pozisyonu
Def Pos VECCMPD     'kapak silindir öğretme pozisyonu
Def Pos VECCMPPS    'tutucu pozisyonu öğretme pozisyonu
Def Pos VECCOC      'silindir oryantasyon öğretme pozisyonu
Def Pos VECCOM      'silindir oryantasyon öğretme pozisyonu
Def Pos VECCTPB     'siyah piston öğretme pozisyonu
Def Pos VECCTPS     'metal piston öğretme pozisyonu
```

'----- vektör atamaları-----

```
VECT30      = (+0.00,+0.00,+0.00,+0.00,+0.00,-45.00)
VECY50      = (+0.00,+50.00,+0.00,+0.00,+0.00,+0.00)
VE CZ2      = (+0.00,+0.00,+2.50,+0.00,+0.00,+0.00)
VE CZ5      = (+0.00,+0.00,+5.00,+0.00,+0.00,+0.00)
VE CZ30     = (+0.00,+0.00,+30.00,+0.00,+0.00,+0.00)
VE CZ50     = (+0.00,+0.00,+50.00,+0.00,+0.00,+0.00)
VE CZ100    = (+0.00,+0.00,+100.00,+0.00,+0.00,+0.00)
VE CZ200    = (+0.00,+0.00,+200.00,+0.00,+0.00,+0.00)
VECCUP      = (+16.25,-30.25,-8.00,+0.00,+0.00,+0.00)
VECCMPZ     = (+56.25,-76.25,-25.00,+0.00,+0.00,+0.00)
VECCOM      = (+14.90,-76.25,-19.00,+0.00,+0.00,+0.00)
VECCOC      = (-16.25,-76.25,-1.10,+0.00,+0.00,+0.00)
VECCOZ      = (+1.35,+0.00,-3.50,+0.00,+0.00,+0.00)
VECCOD      = (+32.50,+0.00,-21.50,+0.00,+0.00,+0.00)
VECCMPPS    = (+56.25,-76.25,-18.00,+0.00,+0.00,-180.00)
VECGA       = (+40.00,+0.00,-2.00,+0.00,+0.00,+0.00)
VE CCTPB    = (+55.00,-12.50,-23.00,+0.00,+0.00,-180.00)
VE CCTPS    = (-55.00,-12.50,-23.00,+0.00,+0.00,+0.00)
VE CZINT     = (+0.00,+0.00,+0.50,+0.00,+0.00,+0.00)
VECMINC     = (+0.00,+0.00,+0.00,+0.00,+0.00,+1.00)
VECCACYL    = (+0.00,+0.00,+0.00,+0.00,+0.00,+72.00)
VECCACOV    = (+0.00,+0.00,+0.00,+0.00,+0.00,+115.00)
VECEP1      = (+0.00,+100.00,+0.00,+0.00,+0.00,+0.00)
VECEP2      = (+0.00,+100.00,+0.00,+0.00,+0.00,+0.00)
,
```

'----- harici adres tanımlamaları-----

```
Def Io B1_R      = Bit,900      'parça siyah değil
Def Io B2_R      = Bit,2        'oryantasyon tanımlaması
```

```

Def Io S1_A      = Bit,3      'başlatma kontrol paneli
Def Io S2_A      = Bit,4      'durdurma kontrol paneli
Def Io S4_A      = Bit,5      'reset kontrol paneli
Def Io PART_AV   = Bit,6      'işlenecek parça hazır
Def Io IP_FI     = Bit,15     'Sürekli istasyon boşluğu
Def Io A1B1_A    = Bit,8      'verici arka yayı
Def Io A1B2_A    = Bit,9      'verici ön yayı
Def Io A2B1_A    = Bit,10     'verici ön kapak
Def Io A2B2_A    = Bit,11     'verici arka kapak
Def Io A1S1_A    = Bit,12     'yay uygun
Def Io B1_A      = Bit,13     'toplama pozisyonunda kapak yok
Def Io B2_A      = Bit,14     'vericide kapak yok
Def Io IP_N_FO   = Bit,15     'istasyon meşgul
Def Io H1_A      = Bit,0      'lamba başlatma kontrol paneli
Def Io H2_A      = Bit,1      'lamba resetleme kontrol paneli
Def Io H3_A      = Bit,2      'lamba Q1 kontrol paneli
Def Io H4_A      = Bit,3      'lamba Q2 kontrol paneli
Def Io A1Y1_A    = Bit,8      'yay ayrılma
Def Io A2Y1_A    = Bit,9      'kapak ayrılma
'
'----- integer tanımlamaları-----

Def Act 1,M_In(4)=0 GoSub *INTSTOP 'durdurmada rutin kesme tanımı
Def Inte CYLTYPE      'silindir tip
Def Inte SLOW         'yavaş hız ayarı
Def Inte FAST         'hızlı hız ayarı
Def Inte SUSLOW       'süper yavaş hız ayarı
Def Inte B2OLD        'oryantasyon ölçüsünün eski değeri
Def Inte B2MESSV      'oryantasyon ölçüsü değeri
Def Inte MAXDEGR      'maximum oryantasyon açısı değeri
Def Inte COSP         'metal piston sayıcısı
Def Inte COBP         'siyah piston sayıcısı
'
'----- palet tanımlamaları-----

Def Plt 2,P5+VECCTPB,P5+VECCTPB+VECEP2,P5+VECCTPB,,5,1,2 'siyah piston
paleti
Def Plt 1,P5+VECCTPS,P5+VECCTPS+VECEP1,P5+VECCTPS,,5,1,2 'metal piston
paleti
'
'----- harici adres tanımlamaları-----

Ovrd 50
SLOW% = 40
SUSLOW% = 40
FAST% = 60
COSP% = 1
COBP% = 1

Dly 0.1
HOpen 1
M_Outw(0) = 0
Spd FAST%
'
'----- mevki tanımlamaları-----

Def Inte PN1
Def Inte PN2
Def Inte PN3
Def Inte PN4

```

```

Def Inte PN5
Def Inte PN6
Def Inte PN7
Def Inte PN8
Def Inte PN9
Def Inte PN10
Def Inte PN11
Def Inte PN12

'----- denetleyici tanımlamaları-----

Def Inte C1
Def Inte C2
Def Inte C3
Def Inte C4
Def Inte C5

'----- geçiş tanımlamaları-----

Def Inte T1_FNS
Def Inte T2_FNS
Def Inte T3_FNS
Def Inte T4_FNS
Def Inte T5_FNS
Def Inte T6_FNS
Def Inte T7_FNS
Def Inte T8_FNS
Def Inte T9_FNS
Def Inte T10_FNS

'----- başlangıç değer atamaları-----

PN1 = 0
PN2 = 0
PN3 = 0
PN4 = 0
PN5 = 0
PN6 = 0
PN7 = 0
PN8 = 0
PN9 = 4
PN10 = 4
PN11 = 10
PN12 = 10

C1 = 1
C2 = 1
C3 = 1
C4 = 1
C5 = 1
Mov P99
*MAINLOOP
*MAINL1

'----- Alt Program Atamaları -----

If PN1 = 1 Then GoSub *TASK1
If PN2 = 1 Then GoSub *TASK2
If PN4 = 1 Then GoSub *TASK3
If PN5 = 1 Then GoSub *TASK4

```

```

If PN6 = 1 Then GoSub *TASK5
If PN7 = 1 Then GoSub *TASK6
If PN8 = 1 Then GoSub *TASK7

'----- T1 geçiři -----

If (PART_AV = 1 ) Then PN1 = 1

'----- T2 geçiři -----

If (PN1=1 and T1_FNS=1 ) Then
PN1 = 0
PN2 = 1
Endif

'----- T3 geçiři -----

If (PN2=1 and T2_FNS=1 ) Then
PN2 = 0
PN3 = 1
Endif

'----- T4 geçiři -----

If (PN3=1 and CYLTYPE% = 0 and PN9>0) Then
PN3 = 0
PN4 = 1
PN9=PN9-1
Endif

'----- T5 geçiři -----

If (PN3=1 and CYLTYPE% = 1 and PN10>0) Then
PN3 = 0
PN5 = 1
PN10=PN10-1
Endif

'----- T6 geçiři -----

If (PN4=1 and T3_FNS = 1 and PN11>0) Then
PN4 = 0
PN6 = 1
PN11=PN11-1
Endif

'----- T7 geçiři -----

If (PN5=1 and T4_FNS = 1 and PN11>0) Then
PN5 = 0
PN6 = 1
PN11=PN11-1
endif

'----- T8 geçiři -----

If (PN6=1 and T5_FNS = 1 and PN12>0) Then
PN6 = 0
PN7 = 1
PN12=PN12-1

```

```

Endif

'----- T9 geçiři -----

If (PN7=1 and T6_FNS = 1) Then
PN7 = 0
PN8 = 1
Endif

'----- T10 geçiři -----

If (PN8=1 and T7_FNS = 1) Then
PN8 = 0
Endif

'----- bařlangıç deęer atamaları-----

T1_FNS = 0
T2_FNS = 0
T3_FNS = 0
T4_FNS = 0
T5_FNS = 0
T6_FNS = 0
T7_FNS = 0
T8_FNS = 0
T9_FNS = 0
T10_FNS = 0

GoTo *MAINL1
End

'-----
'| *PANEL (Alt program) |
'| Robot bařlangıç pozisyonuna g t r l r. |
'|-----

*PANEL
H2_A = 1

*CONT3
If S4_A = 0 Then *CONT3
H2_A = 0
Mov P99, -30
Mov P99
H1_A = 1
*CONT4
If S1_A = 0 Then *CONT4
H1_A = 0
Return

'-----
'| *TASK1 (Alt program) |
'| Silindirin robot tarafından alınarak renk testi alanına bırakılmasıdır |
'|-----

*TASK1

T1_FNS = 0
'
Mov P1+VEZ50

```

```

AUXPOS = P1+VE CZ50
Mov AUXPOS
Spd SLOW%
Mvs AUXPOS - VECZ50
Dly 0.1
HClose 1
Dly 1
Mvs AUXPOS
Spd FAST%
Mov P1+VE CZ50
AUXPOS = VECCUP+VECGA+VE CZ50+P2
Mov AUXPOS + VECZ200
Mov AUXPOS
Mvs AUXPOS - VECZ50
Dly 0.1
HOpen 1
Mvs AUXPOS

T1_FNS = 1
Return

'-----
'|      *TASK2 (Alt program )
'| Silindirin renk testi ve koordinat testi yapılarak birleştirme alanına
'| bırakılmasıdır.
'|-----

*TASK2

T2_FNS = 0
'
AUXPOS = P3 + VECZ50
Mov AUXPOS
Mov P3
Dly 1
CYLTYPE% = B1_R
Mov AUXPOS
If CYLTYPE% = 0 Then GoTo *BLACK1 Else GoTo *NBLACK1
*BLACK1
AUXPOS = VECZ50-VE CZ2+VECCUP+P2
GoTo *CONT1
*NBLACK1
AUXPOS = VECZ50+VECCUP+P2
*CONT1
Mov AUXPOS
Spd SLOW%
Mvs AUXPOS - VECZ50
Dly 0.1
HClose 1
Dly 1
Mvs AUXPOS
Spd FAST%
AUXPOS = AUXPOS - VECCUP + VECCOZ
Mov AUXPOS
Spd SLOW%
AUXPOS = AUXPOS - VECZ50
Mvs AUXPOS
B2OLD% = 0
MAXDEGR% = 0
Dly 0.5

```

```

*FOUND9
B2MESSV% = B2_R
If B2MESSV% = 1 And MAXDEGR% = 0 Then GoTo *LABEL9
If B2MESSV% = 1 And B2OLD% = 0 Then GoTo *FOUND9
*LABEL9
AUXPOS = AUXPOS + VECMINC
Mvs AUXPOS
B2OLD% = B2MESSV%
If MAXDEGR% > 140 Then GoTo *ERRL
MAXDEGR% = MAXDEGR% +1
GoTo *FOUND9
*FOUND9
AUXPOS = AUXPOS+ VECZ50
Mvs AUXPOS
AUXPOS= AUXPOS+VECCOM+VECCACYL
Mov AUXPOS
Mvs AUXPOS-VECZ50
Dly 0.1
HOpen 1
Mvs AUXPOS
Spd FAST%
Mov AUXPOS + VECZ200
'
T2_FNS = 1
Return

```

```

'-----
'|      *TASK3 (Alt program )
'| Siyah pistonunun robot tarafından alınarak birleştirme alanına
'| bırakılmasıdır.
'-----

```

```

*TASK3

T3_FNS = 0
'
If COBP% < 5 Then *CONT21
COBP% = 1
*CONT21
AUXPOS = Plt 1, COBP%
COBP% = COBP% +1
Mov P5 + VECZ100
Mvs AUXPOS + VECZ50
Spd SLOW%
Mvs AUXPOS
Dly 0.1
HClose 1
Dly 1
Mvs AUXPOS + VECZ50
Spd FAST%
Mov AUXPOS + VECZ100
Mov P5 + VECZ100
Spd FAST%
Mov VECZ200+P2
AUXPOS = VECCMPPS+VECZ50+P2
Mov AUXPOS
If CYLTYPE% = 1 Then GoTo *NBLACK3
AUXPOS = AUXPOS-VECZ2
*NBLACK3
Mov AUXPOS

```

```

Spd SLOW%
Mvs AUXPOS - VECZ50
Dly 0.1
HOpen 1
Mvs AUXPOS
Spd FAST%

```

```

T3_FNS = 1
Return

```

```

'-----
'|      *TASK4 (Alt program )
'| Metal pistonun robot tarafından alınarak birleştirme alanına
'| bırakılmasıdır.
'-----

```

```

*TASK4

```

```

T4_FNS = 0
'
If COSP% < 5 Then *CONT22
COSP% = 1
*CONT22
AUXPOS = Plt 2, COSP%
COSP% = COSP% +1
Mov P5 + VECZ100
Mvs AUXPOS + VECZ50
Spd SLOW%
Mvs AUXPOS
Dly 0.1
HClose 1
Dly 1
Mvs AUXPOS + VECZ50
Spd FAST%
Mov AUXPOS + VECZ100
Mov P5 + VECZ100
Spd FAST%
Mov VECZ200+P2
AUXPOS = VECCMPPS+VE CZ50+P2
Mov AUXPOS
If CYLTYPE% = 1 Then GoTo *NBLACK311
AUXPOS = AUXPOS-VE CZ2
*NBLACK311
Mov AUXPOS
Spd SLOW%
Mvs AUXPOS - VECZ50
Dly 0.1
HOpen 1
Mvs AUXPOS
Spd FAST%
'
T4_FNS = 1

Return

```

```

'-----
'|      *TASK5 (Alt program )
'| Yayın robot tarafından alınarak birleştirme alanına bırakılmasıdır.
'-----

```

*TASK5

```
T5_FNS = 0
'
If A1S1_A=1 Then *CONT5
*LOOP1
A1Y1_A=1
*WAIT1
If A1B1_A=0 Then *WAIT1
Dly 1
A1Y1_A=0
Dly 1
*WAIT2
If A1B2_A=0 Then *WAIT2
If A1S1_A=1 Then *CONT5
H3_A = 1
H1_A = 1
*WAIT3
If S1_A = 0 Then *WAIT3
H3_A = 0
H1_A = 0
GoTo *LOOP1
*CONT5
AUXPOS = P4 + VECZ50
Mov AUXPOS
Spd SLOW%
Mvs AUXPOS - VECZ50
Dly 0.1
HClose 1
Dly 1
Mvs AUXPOS
Spd FAST%
AUXPOS = VECCMPPS+VE CZ50+P2
If CYLTYPE% = 1 Then GoTo *NBLACK5
AUXPOS = AUXPOS-VE CZ2
*NBLACK5
Mov AUXPOS
Spd SLOW%
Mvs AUXPOS - VECZ50
Dly 0.1
HOpen 1
Mvs AUXPOS
Spd FAST%
AUXPOS = AUXPOS + VECZ100
Mov AUXPOS
'
T5_FNS = 1

Return
```

```
'-----
'|      *TASK6 (Alt program )
'| Kapağın robot tarafından alınarak koordinat testi yapıldıktan sonra
'| birleştirme alanına bırakılmasıdır.
'|-----
```

```
*TASK6
T6_FNS = 0
'
```

```

If B1_A =0 Then *CONT6
*LOOP2
A2Y1_A = 1
*WAIT4
If A2B1_A=0 Then *WAIT4
Dly 1
A2Y1_A = 0
*WAIT5
If A2B2_A=0 Then *WAIT5
If B1_A = 0 Then *CONT6
H4_A = 1
H1_A = 1
*WAIT6
If S1_A = 0 Then *WAIT6
H4_A = 0
H1_A = 0
GoTo *LOOP2
*CONT6
AUXPOS = P9 + VECZ50
Mov AUXPOS
Spd SLOW%
Mvs AUXPOS - VECZ50
Dly 0.1
HClose 1
Dly 1
Mvs AUXPOS
Spd FAST%
Mov P9 + VECZ200
Mov P9 + VECZ200 - VECY50
AUXPOS = VECZ100+VECCOD+VECGA+P2
Mov AUXPOS
AUXPOS = AUXPOS - VECZ50
Spd SUSLOW%
Mov AUXPOS
Mvs AUXPOS - VECZ50
Dly 0.1
HOpen 1
Mov AUXPOS
Spd FAST%
AUXPOS = VECZ50+VECCOD+P2
Mov AUXPOS
Spd SLOW%
AUXPOS =AUXPOS-VE CZ50
Mvs AUXPOS
Dly 0.1
HClose 1
Dly 1
B2OLD% = 0
MAXDEGR% = 0
Dly 0.5
*NFOUND91
B2MESSV% = B2_R
If B2MESSV% = 1 And MAXDEGR% = 0 Then GoTo *LABEL91
If B2MESSV% = 1 And B2OLD% = 0 Then GoTo *FOUND91
*LABEL91
AUXPOS = AUXPOS + VECMINC
Mvs AUXPOS
B2OLD% = B2MESSV%
If MAXDEGR% > 140 Then GoTo *ERRL
MAXDEGR% = MAXDEGR% +1

```

```

GoTo *NFOUND91
*FOUND91
Mvs AUXPOS+VEZ50
AUXPOS = AUXPOS+VECCOC+VECCACOV+VEZ50
If CYLTYPE% = 1 Then *NBLACK7
AUXPOS = AUXPOS-VEZ2
*NBLACK7
Mov AUXPOS
Spd SLOW%
Mvs AUXPOS - VEZ50
Spd SUSLOW%
Mvs AUXPOS - VEZ50 + VECT30 + VECT30
Spd SLOW%
Dly 0.1
HOpen 1
Mvs AUXPOS
Spd FAST%
'
T6_FNS = 1

Return

'-----
'|      *TASK 7 (Alt program )
'| Montajı tamamlanan parçanın robot tarafından alınarak birleştirilmiş
'| ürün rayına bırakılmasıdır.
'|-----

*TASK7

T7_FNS = 0
'
*WAIT7
If IP_FI = 0 Then *WAIT7
Spd SLOW%
AUXPOS = VECCMPZ+VEZ50+P2
Mov AUXPOS
Mvs AUXPOS - VEZ50
Dly 0.1
HClose 1
Dly 1
Mvs AUXPOS - VEZ50
Mvs AUXPOS
Spd FAST%
AUXPOS = P6 - VEZ50
Mov AUXPOS
Mvs P6
Dly 0.1
HOpen 1
Mvs AUXPOS
Mov P99
IP_N_FO = 0
'
T7_FNS = 1
Return

'-----
'|      *INTSTOP (Alt program )
'| Herhangi bir arıza anında robotun durarak arıza onarımı
'| yapıldıktan sonra lamba start panel butonuna basılarak işleme

```

```
'| kalınan yerden devam edilmesini sağlar. |  
'-----  
  
*INTSTOP  
  
H1_A =1  
*WAIT8  
If S1_A=0 Then *WAIT8  
H1_A=0  
Return 0  
'  
  
*ERRL  
Hlt
```

5. SONUÇ

Tez kapsamında Gaziosmanpaşa Üniversitesi Mekatronik Mühendisliği Bölümü Robotik Laboratuvarında bulunan MITSUBISHI RV-2AJ robotu içeren Robotik montaj istasyonu kullanılarak Robotik Montaj Sistemlerinin Petri Ağı Temelli Kontrolü gerçekleştirilmiştir. Endüstride sıklıkla karşımıza çıkmakta olan Robotik sistemler için Petri ağları kullanılarak denetleyiciler formal olarak hesaplanabilmektedir. Bu tez kapsamında da Petri ağları temelli mevki değişmezlerini kullanan bir yöntem ile gözeticiler hesaplanmaktadır. Hesaplanan gözeticilerin Robot denetleyicisi programlama dili olan Melfa Basic ile gerçekleştirilmesine yönelik genel bir yöntem tez kapsamında önerilmektedir. Deneysel çalışmalar elde edilen yöntem kullanılarak gerçekleştirilen gözeticilerin sistemi başarılı biçimde kontrol ettiği gözlemlenmiştir. Petri ağlarının Melfa Basic dili ile gerçekleştirilmesi ve robotik montaj sistemlerinde kullanımı açısından tez literatüre katkı sağlamaktadır.

KAYNAKLAR

- Arıkan, A. S., 2013, Endüstriyel Robotlar ve Üretimde Uygulama Alanları, MMO Seminer Notları 2013, <http://arsiv.mmo.org.tr/pdf/10948.pdf>. (03.06.2015).
- Balemi, S., Hoffmann, G. J., Gyugyi, P., Wongto, H. and Franklin, G. F., 1993, Supervisory Control of a Rapid Thermal Multiprocessor, IEEE Tran. on Automatic Control, 38, 7, 1040-1059.
- Caloini A., Magnani G.A., and Pezz`e M., 1998, A Technique for Designing Robotic Control Systems Based on Petri Nets, IEEE Transactions on control systems technology.
- Cassandras, C. G. and Lafortune, S., 1999, Introduction to Discrete Event Systems, Kluwer Academic Publishers.
- Chen, S., Ke, J. and Chang, J., 1990, Knowledge Representation Using Fuzzy Petri Nets, IEEE Trans. on Knowledge and Data Eng., 2, 3, 311 – 319.
- David, R. and Alla, H., 1994, Petri Nets for Modelling of Dynamic Systems: A Survey, Automatica, 30, 2, 175-202.
- David, R. and Alla, H., 2005, Discrete, Continuous and Hybrid Petri Nets 30-70.
- Desrochers, A. A. and Al-Jaar, R.Y., 1995, Applications of Petri Nets in Manufacturing Systems, Modelling, Control and Performance Analysis, IEEE Press, New York.
- Filipescu A., Petrea G., Filipescu A., Filipescu S., 2014, Modeling and Control of a Mechatronics System Served by a Mobile Platform Equipped with Manipulator Proceedings of the 33rd Chinese Control Conference July 28-30, Nanjing, China
- Garg, M.L., Ahson, S.I. and Gupta, P.V., 1991, A Fuzzy Petri Net for Knowledge Representation and Reasoning, Information Processing Letters, 39, 165 – 171.
- Gelen G., Aksebzeci B. H., Koç İ. B., ve Uzam M., 2007, Basit Otomasyon Petri Ağları ile Ayrık Olay Sistemlerinin Modellenmesi, sayı 129, sayfa 16-21, Endüstri & Otomasyon.
- Gelen G., Uzam M., 2012, Ayrık Olay Sistemleri için Petri Ağları ve R&W Metodu Tabanlı Modüler Melez Gözeticilerin Sentezlenmesi, Otomatik Kontrol Ulusal Toplantısı Bildiriler Kitabı, TOK 2012, Niğde, Cilt 1, s. 200-205.
- Genrish H. and Lautenbach K., 1981, System Modelling with High Level Petri Nets, Theoretical Computer Science, 13, 109 – 136.

Guia A., DiCesare F., 1991, Supervisory Design Using Petri Nets, Proc. Of the 30 Conf. On Decision and Control, pp. 92-97, Brighton- England.

Guia, A., 1996, Petri Net Techniques for Supervisory Control of Discrete Event Systems, 1st Int. Workshop on Manufacturing and Petri Nets, Osaka - JAPAN, 1 – 21.

Jensen K., 1981, Coloured Petri Nets and the Invariant Method, Theoretical Computer Science, 14, 317-336.

Lauzon, S. C., Ma, A. K. L., Mills, J. K. and Benhabib, B., 1996, Application of Discrete Event System Theory to Flexible Manufacturing, IEEE Control Systems Magazine, 16-1, 41-48,

Lee J-K, Lee T-E, 2002, Automata-based supervisory control logic design for a multi-robot assembly cell, 319–334

Looney, C.G., 1988, Fuzzy Petri Nets for Rule-Based Decision Making, IEEE Transactions on Systems, Man, and Cybernetics, 18, 1, 178-183.

Means of Petri Nets with Imprecise Markings, IEEE Int. Symposium on Intelligent Control, 233 - 237.

Moody J.O., Antsaklis P.J. and Lemmon M.D., 1995, Automated Design of a Petri Net Feedback Controller for a Robotic Assembly Cell I NRA/IEEE Conference on Emerging Technologies and Factory Automation, Paris, France, October 10-13.

Moody J.O., Antsaklis P.J., 1998, Supervisory Control of Discrete Event Systems Using Petri Nets, 22-30.

Morales R.A.G., Miyagi P.E., Riascos L.A.M., 2007, Diagnosis and Treatment of faults Manufacturing Systems on Bayesian Networks and Petri Nets, 19th International Congress of Mechanical Engineering November 5-9, Brasília COBEM

Murata T., 1989, Petri Nets: Properties, Analysis and Application, Proc. of IEEE Vol 77 541-580.

Murata T., 1989, Petri Nets: Properties, Analysis and Applications, Proceedings of the IEEE

Petri, C. A., 1963, Kommunikation mit Automaten Schriften des Rheinisch Westfalischen Inst. für Instrumentelle Mathematik und der Universität Bonn, 1962, Çeviri Green C. F., Applied Data Research Inc., Suppl 1 to Tech report RAD-TR-65-337, NY.

Ramadge, P. J. and Wonham W. M., 1989, The Control of Discrete Event Systems, Proc. IEEE, 77, 1, 81-98.

Reising, W., 1985, Petri Nets : An Introduction, Springer-Verlag.

Rosell J., 2004, Assembly and Task Planning Using Petri Nets: A Survey* Institute of Industrial and Control Engineering, Technical University of Catalonia, Barcelona, SPAIN

Sibertin-Blanc C., 1985, High Level Petri Nets with Data Structures, 6th European Workshop on Applications and Theory of Petri Nets, Helsinki, Finland.

Sreenivas, R.S., 1994, On Asymptotically Efficient Solutions for a Class of Supervisory Control Problems, IEEE Int. Conf. on Systems, Man and Cybernetics, San Antonio, Texas.

Suzuki, I. and Lu, H., 1989, Temporal Petri Nets and Their Application to Modelling and Analysis of a Handshake Daisy Chain Arbiter, IEEE Trans. on Computers, 38, 5,696 – 704.

Uzam M., Gelen G., Koç İ. B., ve Aksebzeci B. H., 2007, Ayrık Olay Kontrol Sistemlerinin Petri Ağları ile Tasarımı ve PLC ile Gerçekleştirilmesi, sayı 125, sayfa 18-22, Endüstri & Otomasyon.

Valette, R., Cardoso, J. and Dubois, D., 1989, Monitoring Manufacturing Systems by Wikipedia, 2003. <http://tr.wikipedia.org/wiki/Robot> (03.06.2015).

www.festo.com, 03.06.2015 tarihinde aktif.

www.mitsubishielectric.com, 03. 06. 2015 tarihinde aktif.

Yasuda G., 2012, Implementation of Distributed Control Architecture for Multiple Robot Systems Using Petri Nets, 75-93

Zhou, M. and Venkatesh, K., 1999, Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach, Intelligent control and intelligent automation, vol. 6, world scientific.

Zhou, M.C. and DiCesare, F., 1993, Petri Net Synthesis for Discrete Event Control of Manufacturing Systems, Kluwer Academic Publishers, Boston.

Ziparo V.A., Iocchi L., Nardi D., Palamara P.F., Costelha H., Ariosto V., 1996, A Formal Model for Representation and Execution of Multi-Robot Plans 25, I-00185 Rome, Italy

Zurawski R. and Zhou M.C., 1994, Petri Nets and Industrial Applications: A Tutorial, IEEE Trans. on Industrial Electronics, 41, 6, 567-583.

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Yasemin İÇMEZ
Doğum Yeri ve Yılı : Yerköy / 1989
Yabancı Dili : İngilizce
Medeni Hali : Evli
TC Kimlik No : 13004344650
Telefon : 0542 640 15 77
e-mail : yicmez@hotmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Lise	Şehit Sedat Nezih Özok Lisesi	2006
Lisans	Karadeniz Teknik Üniversitesi	2012
Yüksek Lisans	Gaziosmanpaşa Üniversitesi	2015

Tecrübe

Staj	TUSAŞ Türk Havacılık ve Uzay San. A.Ş.	08/2010 - 09/2010
Staj	KTÜ Elektrik Elektronik Müh. Böl. DSP Laboratuvarı	07/2011 - 09/2011
İş	Bilecik Kuruluş Yapı Denetim LTD. ŞTİ	10/2012 - 07/2013
İş	Tokat Reyyan Yapı Denetim LTD. ŞTİ	07/2013 - Devam ediyor
İş	Tokat Teknik Bilimler MYO	09/2013 - 02/ 2014

Yayınlar

Y. İÇMEZ, G. GELEN “Elektropnömatik Sistemlerin Kontrolünde Petri Ağlarının Kullanımı ve Mikrodenetleyici Temelli PLC ile Gerçekleştirilmesi” TOK 2014, 11-13 Eylül, Kocaeli

Bilgisayar Programları

C #, C++, Microsoft Office 2007, Excel, Word, PowerPoint, PIC Programlama, PIC-PLC Programlama, Melfa Basic Robotik Programlama.

Sertifikalar

- Atı Eğitim Kurumları Liderlik Eğitimi Sertifikası
- C # Kursu
- KTÜ Kariyer Gelişim Günleri Katılım Sertifikası