T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

RAYLI ULAŞIM SİSTEMLERİNDE ANKLAŞMAN TABLOLARININ DOĞRULUĞUNUN MODEL KONTROLÜ YÖNTEMİYLE TEST EDİLMESİ

BASRİ TUĞCAN ÇELEBİ

YÜKSEK LİSANS TEZİ
KONTROL VE OTOMASYON MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN
YRD. DOÇ. DR. ÖZGÜR TURAY KAYMAKÇI

İSTANBUL, 2016

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

RAYLI ULAŞIM SİSTEMLERİNDE ANKLAŞMAN TABLOLARININ DOĞRULUĞUNUN MODEL KONTROLÜ YÖNTEMİYLE TEST EDİLMESİ

BASRİ TUĞCAN ÇELEBİ tarafından hazırlanan tez çalışması 23/05/2016 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Kontrol ve Otomasyon Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Yrd. Doç. Dr. Özgür Turay KAYMAKÇI

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Yrd. Doç. Dr. Özgür Turay KAYMAKÇI

Yıldız Teknik Üniversitesi

Prof. Dr. Galip CANSEVER

Yıldız Teknik Üniversitesi

Prof. Dr. M. Turan SÖYLEMEZ

İstanbul Teknik Üniversitesi

ÖNSÖZ

Bu çalışmanın yapılması aşamasında desteğini esirgemeyen danışmanım Yrd. Doç. Dr. Özgür Turay Kaymakçı ve her zaman arkamda olan aileme teşekkür ederim.

Mayıs, 2016

Basri Tuğcan ÇELEBİ

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

$\neg$	Bool Cebri – Değil lojik bağlacı
$\wedge$	Bool Cebri – Ve lojik bağlacı
$\vee$	Bool Cebri – Veya lojik bağlacı
$\in$	Elemanıdır
:=	Değer atama
$\cup$	Birleşim
$\cap$	Kesişim
S_i	i adımında durumlar kümesi
$S \models f$	s durumu f zamansal lojik ifadesini sağlar

KISALTMA LİSTESİ

ASM	Abstract State Machine
ATP	Automatic Train Protection
CENELEC	European Committee for Electrotechnical Standardization
CPN	Coloured Petrinet
CSP	Communicating Sequential Processes
CTL	Computational Tree Logic
GDL	Geographic Data Language
LTL	Linear Temporal Logic

ŞEKİL LİSTESİ

ÇİZELGE LİSTESİ

Sayfa

ÖZET

RAYLI ULAŞIM SİSTEMLERİNDE ANKLAŞMAN TABLOLARININ DOĞRULUĞUNUN MODEL KONTROL YÖNTEMİYLE TEST EDİLMESİ

Basri Tuğcan ÇELEBİ

Kontrol ve Otomasyon Mühendisliği Bölümü

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Özgür Turay KAYMAKÇI

Raylı ulaşım sistemleri olası risklere karşı olan duyarlılığı en aza indirgeyebilmek için kullanılacak ürün, yöntem ve tekniklerin uluslararası standartlar tarafından belirlendiği kritik bir sektördür. İlgili standartlar yüksek bir emniyet seviyesine ulaşılabilmesi için geliştirme sürecinde bazı yöntemlerin kullanılmasını şiddetle tavsiye etmektedir. CENELEC 50128, Tablo A.17'de sistem modellenme aşamasında sonlu durum makinalarını ve Tablo A.5'de geliştirilen kontrol algoritmalarının doğrulama ve test aşamasında formal ispat yöntemlerinin kullanılmasını şiddetle tavsiye etmektedir. Bu kapsamda bu çalışmada sinyalizasyon sistemlerinin en kritik bölümü olan anklaşman sistemi incelenmiştir öyle ki anklaşman sistemleri geliştirilirken kodun üretilme aşamasında referans alınan anklaşman tablolarının ASM(Abstract State Machine) ile modellenmesi ve NuSMV yardımıyla model kontrolü gerçekleştirilerek doğruluğu test edilmesi ele alınmıştır.

Bunların yanında anklaşman sistemlerinde otomatik olarak model kontrolünü yapılabilmesi için geliştirilen ASM modeli ve NuSMV kodu genelleştirilmiş bir yapıda oluşturulmuştur.

Tasarlanan yazılım İstanbul Ulaşım A.Ş. tarafından işletilen, T4 Topkapı-Habibler hattı üzerinde bulunan “50. Yıl-Bastabya” istasyonuna ait topoloji ele alınarak düzenlenmiş ve başarılı sonuçlar elde edilmiştir.

Anahtar Kelimeler: anklaşman tablosu, model kontrolü, ASM

ABSTRACT

VERIFYING THE ACCURACY OF INTERLOCKING TABLES FOR SIGNALLING SYSTEMS USING MODEL CHECKING METHOD

Basri Tuğcan ÇELEBİ

Department of Control and Automation Engineering

MSc. Thesis

Adviser: Assoc. Dr. Özgür Turay KAYMAKÇI

Railway transportation systems is a critical sector where products, methods and techniques to be used are identified by international standards so that susceptibility to possible risks can be reduced to minimum level. CENELEC 50128 strongly recommends the utilization of finite state machines during system modeling stage and of formal proof methods during the verification and testing stages of control algorithms. This study examines the most critical part of urban signaling systems, namely the interlocking system. It handles the F of interlocking tables, which occupy a crucial role in the production of the code, through ASM (Abstract State Machine) and testing the verification thereof by conducting model checking through NuSMV.

In addition, ASM model and NuSMV code, developed for the purpose of performing model control automatically in interlocking systems, has been formed in a generalized structure. Consistency of the developed model has further been supervised through fault injection method.

The software design has been arranged based on the topology of “50. Yıl Bastabya” station, operated by Istanbul Transportation Co.

Key words: Interlocking; model checking; Abstract State Machine

BÖLÜM 1

GİRİŞ

1.1 Tezin Amacı

Raylı ulaşım sistemlerinin kullanımı ve yeni raylı sistemlerine olan ihtiyaç ülkemizde ve dünyada giderek artmaktadır. Bu gelişim teorik ve pratik çalışmalarla desteklenmelidir. Yürütülen projede raylı ulaşım sistemlerinde anklaşman tablolarının doğruluğu model kontrolü yöntemi ile kontrol edilerek ülkemizdeki gelişimin desteklenmesi hedeflenmektedir.

1.2 Literatür Çalışması

Raylı ulaşım sistemlerinde yapılan hatalar önemli ekonomik ve can kayıplarına neden olmuştur[1][2][3][4]. Bu yüzden raylı ulaşım sistemlerinin hatada güvenlikli sistemler olması elzemdir. Anklaşman tablolarının doğruluğu ise güvenli operasyonun önemli bir unsurudur. Anklaşman tabloları, trenler için açılabilecek farklı rotalar için gerçekleşmesi gereken şartları ve sistemdeki elemanların durumlarının nasıl değişmesi gerektiğinin belirlendiği listedir. Karmaşık topolojiler için anklaşman tablolarını oluşturmak zorlaşmakta ve hatayı daha olası kılmaktadır. Literatürde konvansiyonel raylı ulaşım sistemleri için anklaşman tablolarının otomatik olarak tasarlanmasını sağlayan yazılımların geliştirildiği görülmektedir[5]. Fakat tasarlanan anklaşman sistemlerinin doğruluğu yapılacak kontrollerle garanti edilmesi gerekmektedir. Model kontrolü sayesinde tasarlanan anklaşman sistemlerinin doğruluğu kontrol edilebilir. Model kontrolü formel bir yöntem olması açısından güvenilirliği yüksek bir kontrol yöntemidir. Demiryollarının güvenli olabilmesi için yazılım geliştirme gereksinimlerinin

anlatıldığı, CENELEC standartlarından EN50128 modülü referans alınır. Bu modülde, demiryollarının güvenlik seviyesi SIL4 olarak belirtilmekte ve yazılım geliştirmede model kontrolü yüksek derecede önerilmektedir.

1.3 Hipotez

Raylı ulaşım sistemlerinin hatada güvenlikli olabilmesi için trafik kontrolünde önemli yer tutan anklaşman tablolarının doğruluğu elzemdir. Anklaşman tablolarının hatada güvenlikli olması için insan hatasının minimuma indirgenmesi gerekmektedir. Yapılan çalışmayla anklaşman tablolarının doğruluğu önerilen modelleme yöntemleri ve önerilen NuSMV dönüşüm algoritmasıyla yapılan model kontrolü sonucu teste tabi tutulabilecektir.

BÖLÜM *2*

RAYLI ULAŞIM SİSTEMLERİ

Bu bölümde raylı ulaşım sistemlerine aşinalığın sağlanması amacıyla raylı ulaşım sistemleri hakkında genel bilgi verilmiştir. Raylı ulaşım sistemlerinin geçmişi ve gelişimi anlatılmış ardından da günümüzde kullanılan raylı ulaşım sistemlerinde kullanılan operasyon presiplerinden bahsedilmiştir. Çalışmanın konusu anklaşman tablolarında model kontrolünün gerçekleştirilmesi olduğu için özellikle aklaşman prensibi daha ayrıntılı olarak incelenmiştir.

2.1 Raylı Ulaşımın Gelişimi

Temel olarak raylı ulaşım, sabit olarak hazırlanmış veya döşenmiş bir yol üzerinde yani raylar üzerinde hareketin sağlandığı ulaşım tipidir. İlk zamanlarda taş yolların oyulması ile oluşturulan raylar Yunanistan ve Roma'da raylı ulaşımın izlerini göstermektedir[26]. Daha sonrasında ise uzun bir süre unutulmuş olan raylı sistemler Avrupa'da aydınlanma çağıyla birlikte neredeyse her madende kullanılmıştır. O dönemlerde çoğunlukla hayvan gücüyle hareketin sağlandığı raylı ulaşım endüstriyel çağla birlikte yerini buhar motoru ile çalışan araçlara bırakmıştır. Buhar motoru raylı ulaşım sistemleri için bir devrim olmuş ve bu sektörün gelişmesini sağlamıştır. İlk olarak malzeme taşıma amaçlı kullanılan raylı ulaşıma sonrasında toplu taşıma da eklenmiş ve bu sayede raylı ulaşımın kullanım alanı genişlemiştir.

Şekil 2. 1 Diolkos : Eski Yunanistan'da raylı ulaşım izleri

Elektriğin yaygınlaşmasıyla beraber raylı ulaşımda da elektrik enerjisi kullanılmaya başlanmış ve daha çok şehir içi toplu ulaşımda elektrikli tramvaylar raylı ulaşımda yerini almıştır. Ayrıca elektrik ve dizel motorlarındaki gelişmeler raylı ulaşımda daha fazla yük taşınmasını sağlamış ve raylı ulaşımın kullanım alanını daha da genişletmiştir. Bu gelişmelerin yanında ikinci dünya savaşı sonrası işçilik maliyetlerinin artmasıyla buharlı trenlerin işletilmesi ekonomik olmaktan çıkmış ve bu yüzden çoğu buharlı tren yerini dizel motorlu trenlere bırakmıştır.

Raylı ulaşımda güvenlik ilk zamanlarda araçların fazla hızlı olmaması ve kullanımının çok yaygınlaşmaması nedeniyle pek önemsenmemiştir. Kullanımın yaygınlaşması ve tren hızlarının artmasıyla trenler arasındaki mesafeler ve tren hareketlerinin senkronizasyonu önem kazanmıştır. Bu yüzden öncelikle belirli aralıklarla trenlerin hareketlerini denetleyen ve düzenleyen görevliler kullanılmıştır. Bunun sonucu olarak ilk sinyalizasyon sistemleri ortaya çıkmıştır. Fakat işaretçilerin makinistler tarafından görülmesinin zor olması ve fren mesafesinin ayarlanmasındaki zorluklar farklı yaklaşımların geliştirilmesini gerekli kılmıştır.

İlk olarak trenlerin belli zaman aralıklarıyla hareketi sağlanmış fakat diğer trenlerin hareketlerinin tam olarak bilinememesi zaman aralık yönteminin yerini mesafe aralık yöntemine bırakmasına neden olmuştur. Bununla birlikte demiryolu bloklara bölünerek her bloğun girişini kontrol eden işaretçiler konulmuştur. Bu işaretçiler sayesinde makinistler girecekleri bloklar hakkında bilgi edinme şansını bulmuştur. Ayrıca telgraf teknolojisi sayesinde istasyonlar arasındaki haberleşme de mümkün kılınarak trenin gideceği istasyonda gereken işlemlerin uygulanması sağlanmıştır.

On dokuzuncu yüzyılın ortalarında işaretçilerin ve makasların durumlarının senkronizasyonunu sağlamak amacıyla sinyal kutuları kullanılmaya başlanmıştır. Bunun amacı sinyalizasyonun ve makas durumlarının belli kurallara göre kilitlenmesi ve kurallara uymayan durumlara izin verilmemesini sağlamak olmuştur. Anklaşman sisteminin temelleri bu şekilde atılmıştır.

Kontrol sistemlerindeki gelişmelerle birlikte sinyalizasyon dolayısıyla tren hareketleri lojik devrelerle kontrol edilmeye başlanmıştır. Bunun için kontrol merkezlerinde büyük röle odaları kullanılmıştır. Elektronik ve haberleşme teknolojisindeki gelişmelerle de şu anda kullanılan kontrol sistemleri oluşturulmuş ve trenlerin hareketleri haberleşme ağları üzerinden kontrol edilip, gelişmiş kontrol algoritmaları ile trenlerde makinistlerin kontrolü kısıtlanmış, daha hızlı tepki verebilen elektronik kontrolörlerle trenlerin kontrolü gerçekleştirilmeye başlanmıştır.

2.2 Raylı Ulaşım Sistemlerinde Operasyon Prensipleri

Bu bölümde çalışmanın daha iyi anlaşılması ve anklaşman sistemleri hakkında genel bir bilgi birikimi oluşturmak adına raylı ulaşım sistemlerinin operasyon prensiplerinden bahsedilecektir. Daha ayrıntılı bilgi için G. Theeg ve S. Vlasenko'nun editörlüğünü yapmış olduğu "Railway and Signal Interlocking" çalışması incelenbilir[27].

Endüstriyel devrimle birlikte, raylı ulaşım sistemleri önemli hale gelmiş ve dolaysıyla kullanımında artmıştır. Raylı ulaşım sistemlerinin kullanımının yaygınlaşması ile işletim operasyon prensipleri gelişmeye başlamıştır. Operasyon prensipleri ülkeden ülkeye değişiklik göstermektedir. Dünyada genel olarak kullanılan operasyon prensipleri aşağıdaki gibi sıralanabilir:

- Britanya Operasyon Prensibi
- Alman Operasyon Prensibi
- Kuzey Amerika Operasyon Prensibi

Bunun dışında Rusya, İskandinavya, Batı Avrupa gibi bölgelerde bahsedilen genel sistemlerin bir karışımı olarak kullanılan operasyon sistemleri kullanılmaktadır.

Operasyon prensiplerinin farklılaşması 19. yüzyıla kadar uzanır. Raylı ulaşımın ilk kullanılmasından 1870'lere kadar raylı ulaşım deneme yanılma ile geliştirilmiş ve şimdiye kadar kullanımın sürdüren bir çok operasyon prensibi oluşturulmuştur. Sinyalizasyon sistemleri ile Avrupa sistemleri Amerikan prensipleri geniş bir ölçüde farklılaşmış ve Alman prensipleri de Avrupa ile geniş ölçüde farklılık göstermeye başlamıştır.

2.2.1 Ray Bölgelerinin, İstasyonların ve Sinyallerin Sınıflandırılması

Operasyonel kullanım açısından raylar genel olarak iki sınıfa ayrılmaktadır. Bu rayların kullanımı ve tanılanması değişkenlik gösterse de prensipte aynı kural geçerlidir. Trenlerin hareket için kullandığı raylar "Ana Ray Bölgeleri", park veya bakım için kullanılan raylar "İkincil Ray Bölgeleri" olarak adlandırılmaktadır. Ana ray bölgeleri üzerinde ve makaslarda genel olarak sinyalizasyon sistemleri bulunmaktadır. İstasyonlar ana ray bölgelerinden ve ikincil ray bölgelerinden oluşmaktadır.

2.2.2 Sinyallerin kullanımı

Sinyaller genel olarak trenin bir ray bölgesine giriş yapıp yapmamasını kontrol eder. Trenin bir ray bölgesini girişini önlemek için kullanılan sinyallere "Ana Sinyaller" denir. Bu sinyalden önce trenin fren mesafesi tahmin edilerek "Yardımcı Sinyaller" kullanılır. Bu sinyallerin yanı sıra manevralar için uyarıda bulunması için sinyaller de kullanılmaktadır. "Manevra Sinyalleri" kullanılırken ana sinyaller muhafaza edilir. Bazı operasyon prensiplerinde manevra sinyalleri ile ana sinyaller bileşik olarak kullanılır(Kuzey Amerika Operasyon Prensipleri).

Sinyaller operasyonel açıdan üç ana başlık altında incelebilirler:

- Kontrollü Sinyaller
- Yarı-Otomatik Kontrollü Sinyaller
- Otomatik Kontrollü Sinyaller

Kontrollü sinyaller sadece operatörler tarafından değiştirilen sinyallerdir. Yarı-Otomatik sinyaller ise operatör kontrolü veya otomatik modu seçilebilen sinyallerdir. Otomatik mod ise merkezi bir sistem tarafından otomatik olarak kontrol edilen sinyaller olarak sınıflandırılırlar.

2.2.3 İstasyon ve Anklaşman Alanlarının Tanımı

Zaman çizelgesi operasyonu bakış açısıyla istasyon, zaman çizelgesinin işlendiği yer olarak belirtilebilir. Operasyonel açıdan bakıldığında trenlerin durduğu, yükleme veya boşaltma yapıldığı veya yolcuların iniş-biniş yaptığı yerler olarak tanımlanabilir.

Anklaşman trenin bir noktadan diğer noktaya süreğen ve güvenli bir şekilde ulaşması için sinyalizasyon ve makas geçiş noktalarının uygun durumlarda ayarlanmasıdır. Sinyalizasyon veya makaslı ana ray bölgeleri genel olarak anklaşman bölgeleridir. Bu bölgelerde operasyon prensibine göre sinyallerin ve makasların durumları, bölgede lokal olarak veya uzakta kontrol merkezi tarafından kontrol edilebilir. İstasyon içinde bulunan ray bloklarına “İstasyon Ray Blokları”, istasyon dışında bulunanlar ise sadece “Ray Bloğu” olarak adlandırılmaktadır.

Anklaşman bölgeleri farklı operasyonları için farklı çizilir. Kuzey Amerika Operasyon Prensiplerine göre anklaşman bölgeleri makasların giriş ve çıkışı ile sınırlandırılır. Buradaki sinyaller ardışık anklaşman bölgelerine göre ayarlanmaktadır. Britanya Operasyon Prensiplerine göre bütün istasyon anklaşman alanı olarak kabul edilmektedir. Anklaşman alanında kullanılan sinyallere anklaşman sinyalleri denmektedir.

Anklaşman sinyallerinin sınıflandırılması kullanılan operasyon prensibine göre değişse de genel olarak aşağıdaki gibidir:

İstasyon Giriş Sinyalleri: Bir ray bloğundan istasyon ray bloğuna geçişte bulunun anklaşman sinyalleri.

İstasyon Çıkış Sinyalleri: İstasyon ray bloğundan ardışık ray bloğuna bulunan anklaşman sinyalleri.

Ray Bölgesi Sinyalleri: Ray bloğundan diğer bir ray bloğuna geçişte bulunan anklaşman sinyalleri

Ara Sinyaller: İstasyon bölgesinin içinde bulunan bir istasyon ray bloğundan diğer istasyon ray bloğuna geçişte bulunan anklaşman sinyalleri.

2.2.4 Tren Hareketinin Kontrolü

Trenler arasındaki fren mesafesi düşük hızlar dışında(15km/s-30km/s) görüş mesafesi dışında olmaktadır. Düşük hızlar ise normal operasyonda sadece makas geçişlerinde veya özel koşullarda kullanılmaktadır. Dolayısıyla fren mesafesi genel olarak görüş mesafesi dışında olmaktadır. Bu yüzden trenlerin hareketleri, konumları izlenerek sinyalizasyon sistemleri uygun durumlara getirilir.

Trenlerin hareketleri, trenler arası mesafeler ilk olarak zaman çizelgeleri ile kontrol edilmiştir; fakat 1870 lerden sonra Avrupa da bu kontrol prensibi terk edilmiş, yerini ray blokları ile kontrol almıştır. Amerika da belli bir oranda daha fazla kullanılma fırsatı bulduysa da sonrasında bu yöntem tamamen terk edilmiştir.

Ray devreleri, ray bloklarının oluşturulmasını sağlamıştır. Bu şekilde raylar bloklar halinde gruplanarak trenler arasındaki mesafeler bloklarla belirlenmiştir. Daha sonrasında Otomatik Lokomotif Sinyalizasyon ile ray blokları yerine sürekli mesafe takibi yapılmaktadır. Bu yöntemde tren konumları sürekli bilinmek zorundadır.

2.2.4.1 Sabit Blok Operasyonu

Trenler arasındaki mesafenin bloklar halinde ayarlandığı durumlarda kullanılmaktadır. Ray bloklarına giriş ve çıkış indikasyon ve kontrol sinyalleriyle sınırlandırılır. Her blokta

sadece bir tren bulunabilmektedir. Bloğun trenin girmesi için güvenli olması için aşağıdaki koşullar sağlanmalıdır:

- Öndeki tren bloğu tamamen terk etmiş olmalı.
- Öndeki tren durma sinyali ile bloğa girecek olan trenden korunmalı.
- Tren ters yönlü hareketlerden korunmalıdır.
- Bazı operasyonlarda blok kontrol sinyallerinden sonra belli bir mesafe garantilenmiştir. Böyle uygulamalarda bloğa giriş izni verilebilmesi için bloktan çıkan trenin çıkış sinyalinden sonra bu mesafeyi katetmesi beklenmektedir.

İki tren arasındaki minimum aralık iki tren arasında güvenli operasyon için bırakılan zaman aralığıdır. Bu zaman aralığı,

- Bloğun tamamen terk edilmesi için geçen zaman,
- Blok giriş sinyalinin görme mesafesine girme zamanı,
- Blok giriş sinyaline ulaşmak için geçen zaman,
- Blok giriş ve çıkış sinyalleri arasında geçen zaman,
- Bloğun terk edilmesi için geçen zaman,
- Bloğun bloke sinyalinin kaldırılması için geçen zaman,

bileşenlerinin toplamından oluşmaktadır. Bu zaman aralığı süreğen hareket halindeki trenler için yukarıda belirtildiği gibidir. Eğer planlı duruş yapan trenler söz konusuysa, blok girişinin görme mesafesine girmesi için geçen zaman ve blok giriş sinyaline ulaşmak için geçen zaman sonraki blok içinde gerçekleşmiş sayılır ve hesaba katılmaz.

2.2.4.2 Otomatik Lokomotif Sinyalizasyon

Otomatik Lokomatif Sinyalizasyon sistemi trenin otomatik olarak kontrol edilmesini sağlayan sinyalizasyonun merkezi olarak gerçekleştirildiği sistemlerdir. Bu şekilde operasyonu gerçekleştirilen raylı ulaşım sistemlerinin neredeyse hepsinde Otomatik Tren Koruma (ATP – Automatic Train Protection) Sistemi bulunmaktadır. Fakat bazı sistemlerde otomatik frenleme bulunmaksızın sadece sinyalizasyon olarak

kullanılmaktadır. Bu sistemlerde genelde ray yolunda başka bir sinyalizasyon elemanı bulunmaz; ancak bazı sistemlerde otomatik lokomatif sinyalizasyonu olmadan da operasyonu sürdürebilmek adına kullanılabilmektedir.

Otomatik Lokomotif Sinyalizasyon sistemi ile hem "Sabit Blok Bölgeleri" hem de "Hareketli Blok Bölgeleri" ile tren kontrolü yapılabilmektedir. Sabit ray blokları ile otomatik lokomotif sinyalizasyon sistemi, blok giriş sinyaline yaklaşma süresi ve sinyal izleme süresinin ortadan kaldırılmasıdır Zira sinyal kondüktör tarafından önceden bilinebilmektedir. Hareketli blok bölgeleri ile operasyonun temel farkı ray blokları yerine trenlerin kontrollerinin fren mesafelerine göre yapılmasıdır. Bunun için tam fren mesafesi veya göreli fren mesafesi kullanılabileceği öne sürülmektedir. Teoride güvenli olduğuna inanılsa da hareketli blok bölgeleri akademik çalışmalarda kullanılmakla kalmıştır.

2.3 Anklaşman Prensipleri

Anklaşman sistemi, raylı ulaşım sistemlerinde bilgilerin işlenmesi ve teknik anlamda operasyonun güvenli olarak gerçekleştirilmesi görevini üstlenmektedir. Bunu gerçekleştirmek için anklaşman sistemi çeşitli ekipmanlarla rayların kullanım durumunu ve hareketli ray bölgelerinin(makaslar vb.) konumunu kontrol eder. Anklaşman sistemi temel olarak iki önemli prensibe dayanmaktadır:

- Tren hareketine izin verilebilmesi için ilgili tüm hareketli ray bölgeleri uygun konumda ve kilitlenmiş ve bahsi geçen trenin kullanımında olduğu sürece aynı konumda kilitli olarak kalmalıdır.
- Sabit ray bloğu ile operasyonu için bahis konusu ise, trenin bir ray bloğuna girmesi için ray bölgesinin başka bir tren tarafından kullanılmıyor olması ve ray bloğuna giriş için gerekli güvenlik önlemlerinin alınması gerekir.

Raylı ulaşım sisteminde güvenlik konusunun bu kadar önemli olmasının nedeni ray ve tren arasındaki sürtünmenin düşük olması sonucu fren mesafesinin uzun olması ile ani frenlemenin mümkün olmaması ve trenlerin raylar yani belirli bir güzergahta hareket edebilmesi sonucu raydan çıkma veya süreğen olmayan bir rota düzenlemesinin önüne

geçilmesi gerektiğidir. Bu nedenlerden ötürü raylı ulaşım sistemleri gerekli önlemler ve düzenlemeler yapılmadığı zaman can ve mali kayıplara neden olabilecek, dikkatli kontrol edilmesi gereken tehlikeli bir sistemdir. Bu yüzden aşağıdaki gibi önlemlerin alınması gerekmektedir:

- Trenler hareket yönünde gerekli önlemler alınmalıdır.
- Trenler ters yöndeki tren hareketlerinden korunmalıdır.
- Kanat koruması yapılmalıdır.
- Hareketli ray bölgeleri trenin geçişi için güvenli pozisyonda olmalıdır.
- Ray bölgeleri için güvenli hız limitleri tanımlanmalıdır.
- Hemzemin geçitlerin güvenliği sağlanmalıdır.
- Sistem dışı etkenler için güvenlik önlemleri alınmalıdır.

Anklaşman sistemi bu güvenlik şartlarının sağlanması ve güvenli bir operasyon için tren hareketinin kontrolünü sağlamaktadır.

BÖLÜM 3

MODEL KONTROLÜ

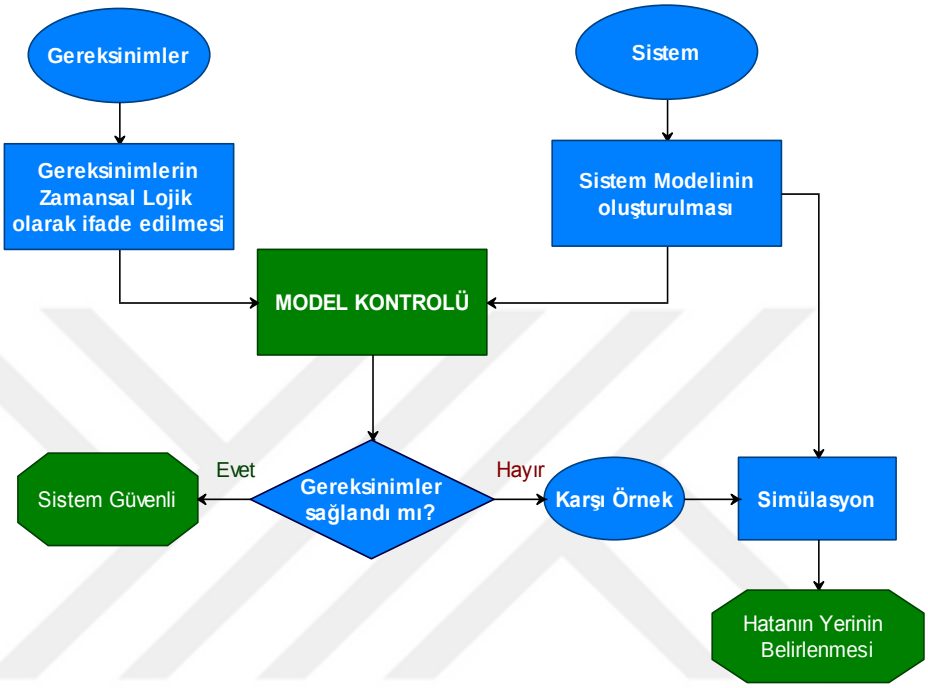
Sistemlerin, devreye alınmadan önce belli testleri geçmesi gerekir. Tasarlama aşamasından sonra geçilen test aşamasında sistemdeki hatalar tespit edilir. Fakat bazı tasarım hataları test esnasında fark edilse bile bazı hataların düzeltilmesi ya fazla zaman almakta ya da telafisi olmamaktadır. Bunun önemli zaman ve ekonomik kayıplara yol açacağı aşikardır. Simulasyon yöntemi tasarım aşamasında test yapılmasını olanaklı kılmasına rağmen simulasyonda olası bütün hataların görülemeyebilr. Bu durumda sistemin tamamen hatada güvenli olup olmadığı anlaşılamayabilir. Model kontrolü yöntemi olası bütün durumların test edilebilmesini sağlayabilmesi ve otomatik olarak kontrol edilebilirlik sunduğu için eğer sistemin davranışına uygun bir model oluşturulmuşsa yapılan test sonucu sistemin hatalı olup olmadığının belirlenmesi için daha uygun olan formal bir yöntemdir.

Model kontrolü Edmund M.Clarke ve Allen Emerson[6] ve bağımsız olarak J.P Quielle ve J.Sifakis[7] tarafından yapılan çalışmalar sonucu ortaya atılmıştır. Model kontrolü kısaca aşağıdaki gibi tanımlanabilir:

M bir Kripke yapısı olsun(durum-geçiş grafı). f ise zamansal lojik olarak ifade edilen bir formül olsun. M yapısındaki $s \models f$ olan bütün *s* durumlarının araştırılmasına model kontrolü denir[8].

Model kontrolünün uygulanması için öncelikle kontrol edilecek sistem uygun bir formel modelle ifade edilmesi gerkmektedir. Modelin uygunluğu, sistemin davranışını tam olarak yansıtabilmesiyle ölçülür. Modelin uygunluğunu test etmek için simulasyon yöntemi uygulanabilir. Model oluşturulduktadan sonra kontrol edilecek olan, sistemin

sağlaması gereken veya hata arz edebilecek özellikler, zamansal lojik formülleri ifade edilerek kontrol gerçeleştirilir. Kontrol sonucu alınan karşı örnek(counter example) ile özelliklerin sağlanıp sağlanmadığı kontrol edilir ve sistemde uygun modifikasyonlar yapılır.



Şekil 3. 1 Model Kontrolü Prosesi

Model kontrolü için bir diğer yöntem sembolik model kontrolü ile ispat yapma esasına dayanır. Model, formüllerden oluşmakatadır ve bazı ispat algoritmaları ile gereksinimlerin durum uzayında bulunup bulunmadığı araştırılır. Fakat model kontrolü için oluşturulmuş yazılım araçları ile kontrolü gerçekleştirmek, değinilen yönteme göre daha az yetkinlik ve zaman gerektiğinden endüstrinin hızlı cevap arayışına daha uygun bulunmuştur.

3.1 Modelleme

Model kontrolü yapılabilmesi için sistemin öncelikle formel olarak ifade edilmesi gereklidir. Formel modeller durum-geçiş diyagramları, petri-net ağları, ASM, CSP ,CPN vb. modelleme yöntemleri ile oluşturulabilir. Model oluşturulurken hangi yöntem

kullanılırsa kullanılsın önemli olan modelin sistemin davranışını yansıtmasıdır. Eğer model uygun bir model değilse yapılacak olan kontrolün herhangi bir anlamı olmayacaktır. Bu yüzden modelleme safhasında sistem hakkında ayrıntılı bilgiye sahip olunmalıdır.

Oluşturulacak olan modelin sistemin davranışını tam olarak yansıtmasının yanında modelin sade olması da önem arz etmektedir. Sistemin elemanları fazlalaştıkça modelde incelenmesi gereken durumların artması söz konusudur ve ulaşılması mümkün durumlar üssel olarak artmaktadır[9]. Çok fazla durum olduğunda, durum patlaması(state explosion) olarak adlandırılan ve model kontrolünün yapılamamasını neden olan sorun ortaya çıkmaktadır. Durum patlaması sorununu gidermek için farklı çözüm öneriler getirilmiştir. Örneğin Kenneth McMillan doktora tezinde bu sorunu çözmek için modelin OBDD(Ordered Binary Decision Diagram) olarak ifade edilmesini önermiştir[10]. Bir başka çalışmada ise karşı örnek cevabı kullanılarak modelin daha küçük bir modele nasıl indirgenebileceği ve indirgenmiş modelin sisteme uygun olması için iyileştirmelerin nasıl yapılması gerektiği açıklanmıştır[11].

Anklaşman tablolarında farklı rota olasılıklarının bulunmasının yanında rotaların kendi içinde de farklı olası durumlar mevcuttur. Bu olasılıkların her biri modelde bazı durumların stokastik olarak değişmesini gerektirmektedir. Karmaşık topolojiler için oluşturulan modellerde durum patlaması sorunuyla karşılaşılabilir. Bu yüzden oluşturulacak olan modelin sade olmasına dikkat edilmelidir.

Anklaşman sistemlerinin modelinin nasıl bir yöntemle oluşturulması gerektiğinin ve hangi model kontrolü yazılımı kullanılmasının daha avantajlı olacağının kararını vermek bu çalışmanın yanıtlaması gereken önemli problemlerden biridir. Literatürde bu konuda çeşitli çalışmalar karşımıza çıkmıştır. Örneğin Winter, modelini CSP(Communicating Sequential Processes)[12] formel modelleme dili ile hazırlamış ve model kontrolünü FDR(Failures-Divergences Refinement) yazılımı ile yapmıştır[13]. CSP modellemeden kısaca bahsetmek yararlı olacaktır. CSP modellemede farklı proseslerin birbiriyle etkileşim içinde bulunması ve haberleşmesi söz konusudur. Fakat istenirse bazı proseslerin birbirinden izole olarak paralel çalışması sağlanabilir. Ayrıca prosesler arası geçişler koşullara bağlı olarak tanımlanabilmektedir. Bu çalışmada tren çarpışması

ve raydan çıkma durumları kontrolü edilmiş fakat CSP dili anklaşman tablolarını modellemek için yetersiz bulunmuştur.

Simpson çalışmasında Britanya Raylı Sistemleri için GDL(Geographic Data Language) notasyonuna uygun şekilde oluşturduğu modeli CSP olarak ifade etmiştir. Model kontrol yazılımı olarak FDR2 yazılımı kullanılmıştır[14].

Başka bir çalışmada ise anklaşman sisteminin modeli PN(Petri-Net) ile oluşturulmuş model kontrolü için TAPAAL model kontrolü yazılımı kullanılmıştır[29][30].

Winter sonraki çalışmasında Robinson ile durum-geçiş diyagramlarına yakın bir dil olan ASM(Abstract State Machine) notasyonu ile model oluşturmuş ve NuSmv ile model kontrolü yapmıştır[15]. ASM notasyonunun[16][24], sistem yapısının ve dinamik özellik kazandaran durumlar arası geçişlerin aynı notasyonla yapılabilmesi açısından CSP ile yapılan modellere göre daha anlaşılır bir şekilde ifade edildiği savunulmuştur. ASM modeli kısaca durumlar ve başlangıç durumu S_0 ile S_i arasındaki geçişi sağlayan geçiş kuralı gruplarından oluşmaktadır. S_{i-1} durumundan S_i durumuna geçişte, t_i geçiş kuralı grubu içindeki geçiş kurallarının hepsi senkron olarak gerçekleştirilir.

$$S_0 \xrightarrow{t_1} S_1 \xrightarrow{t_2} S_1 \ldots \xrightarrow{t_n} S_n$$

Modelde fonksiyonlar statik(değişmeyen), dinamik(değişen) ve harici(etkileşimli) olarak çeşitlendirilmiştir. Topoloji statik fonksiyonlar olarak tanımlanmış ve anklaşman tablosuna göre gerçekleştirilecek olan işlemler dinamik fonksiyonlar olarak ifade edilmiştir. Harici fonksiyonlar,bağımsız olarak değişebilen değişkenler için kullanılmıştır. Ayrıca tren modeli oluşturulup model kontrolünün sonuçlarının algılanması kolaylaştırılmıştır.

Bir başka çalışmada ise model, GCL(Guarded Command Language) notasyonuyla oluşturulup proseseslerin haberleşmesi CSP ile modellenmiştir[17]. Anklaşman sistemlerinin otomatik olarak model kontrolü yapılabilmesi için bir çekirdek program oluşturulmuş, bu şekilde yapısı değişebilen anklaşman sistemleri için ortak bir altyapı hazırlanmıştır. Değişen yapılar eklenerek model kontrolünün yapılması sağlanmıştır.

Anunchai SRT(State Railway of Thailand) formatında oluşturulan anklaşman sisteminin modellenmesinde CPN(Coloured Petrinet) kullanmıştır[18]. Örnek sistem olarak Panthong istasyonu ele alınmıştır. CPN modeller jetonların kendisine has özellikleri olması ile Petri Net modellerden ayrılmaktadır[19]. Bu çalışmada modelin büyümemesi için bazı kabuller yapılmıştır. Örneğin trenlerin uzunluğu ihmal edilmiş, trenlerin sadece bir ray bölgesini kapladıkları düşünülmüştür. Ayrıca tren hızı diğer proseslerden daha yavaş olması ile anklaşman sisteminin tren hareketini denetleyememesi engellenmiştir. Ekipmanların ise her zaman işler halde olduğu düşünülmüştür. Trenlerin aynı anda bir ray bölgesini kapsaması kabulu dışındaki diğer kabullerin makul olduğu görülmektedir. Zira trenin hızının kontrol mekanizmasından hızlı olması gerçeğe uygun değildir. Ayrıca ekipmanların arızalanabilir olduğunu kabul etmek, anklaşman sisteminin doğruluğu kontrol edildiğinden amaca uygun değildir.

Araştırmalar sonucu anklaşman sisteminin modellenmesinde ASM notasyonunun kullanılmasının daha uygun olduğuna karar verilmiştir. ASM diğer formel dillere göre genel kapsamda daha anlaşılır bulunmuştur. Bu şekilde yapısı birbirine benzemeyen anklaşman sistemlerine de model kontrolünün kolayca uygulanabilmesi ve kodun daha anlaşılabilir olması öngörülmüştür. Ayrıca model kontrolü yazılımı olarak literatürde sıkça kullanılmış olan ve incelemeler sonucu ileride yapılabilecek olan otomatik anklaşman sisteminin model kontrolünün gerçekleştirilmesi gibi çalışmalara uygun oluğu anlaşılan NuSMV yazılımının kullanılmasına karar verilmiştir.

NuSMV asenkron ve senkron sistemlerin model kontrolünü sağlamakta olup, modüler ve hiyerarşik tanımlamalara izin vermektedir. Modüler tanımlama anklaşman sisteminde bulunan farklı elemanların ortak yapısının tanımlanıp, bu ortak yapı üzerinde somutlaştırmayı sağlayarak yazılımda kolaylık sunmaktadır. Bunun yanında NuSMV deterministik olmayan değişkenlerin de tanımlanmasını sağlayabilmektedir. Bu şekilde rastsal olarak gelişebilecek olaylar deterministik olmayan değişkenlerle tanımlanabilmektedir.

3.1.1 ASM Notasyonu ve Anlamsal Tanımlamalar

ASM notasyonu endüstriyel sistemlerde, protokollerde, gömülü sistemlerde, yazılım uygulamalarında, kontrol ve doğrulama işlemlerinde kullanılan formel ifade

biçimlerinden biridir. Ayrıca sistemin modellemesi için kullanılan notasyon çoğu model kontrolü araçlarında kullanılan geçiş şartlarına uyum göstermektedir ve model özellikleri kaybolmadan model kontrolü yazılımına çevrilebilmektedir. Bu nedenle sistemin formel olarak ifade edilmesinde ASM notasyonu kullanılmıştır.

ASM durumlar ve geçiş kuralları olmak üzere iki ana başlıkta incelenmelidir. Durumlar kelime hazinesini, fonksiyonların tanımını ve değişkenleri içeren yapılardır. Geçişler ise durumlar arası geçiş kurallarının tanımlandığı yapılardır.

Bu bölümde modelin matematiksel altyapısının oluşturulması açısından ASM notasyonu ve anlamsal tanımlamalar incelenmiştir[24].

3.1.1.1 Durumlar

Durumlar modelin tabanını oluşturulur. Değişken isimleri, fonksiyon isimleri ve tanımlamaları, konum bu bölümde anlatılacaktır.

Tanım 3.1 (***Kelime Hazinesi)*:** Σ kelime hazinesi, $n \in N_0$ argüman sayısına sahip fonksiyon isimlerinin sonlu kümesidir. Aynı zamanda eğer belirtilmemişse her kelime hazinesi *tanımsız, doğru* ve *yanlış* lojik ifadeleri içerir. Ayrıca Bool Cebri'nde tanımlanmış ¬, ^, ˅ operatörleri kelime hazinesine dahildir.

Tanım 3.2 (*Konum*) : Kelime hazinesi ile belirlenerek, argüman sayısı $n \geq 0$ ve $v_1 ... v_n$ argümanlarından oluşan f fonksiyon ismi ile tanımlanmış $(f, (v_1, ..., v_n))$ ikilisine *konum* denir. $v_1 ... v_n$ argümanları herhangi bir tipte tanımlanmış v_i dinamik parametre değerleri listesinden oluşmaktadır.

Tanım 3.3 (*Durum*) : Σ kelime hazinesinde tanımlı bir *A durumu* boş olmayan bir *X kümesi*(A durumunun süper uzayı) için fonksiyon isimlerinin değerlendirmelerini içerir. *Değerlendirme,* $n > 0$ olmak üzere *n* argümanlı f fonksiyonu için $f^A : X^n \to X$ fonksiyonu olarak tanımlanır.

Durumlar bir durumdan diğerine değişen fonksiyonların değerlendirmelerine bağlıdır. Fonksiyonlar birbirleri ile ilişkilerine ve değerlendirmelerine göre farklı gruplara ayrılır:

Statik Fonksiyon: Fonksiyonun değerlendirmesi değişmez. Kısaca sabit değerli fonksiyonlar olarak tanımlanabilir.

Dinamik Fonksiyon: Modele bağlı olarak değerlendirmesi değişen fonksiyonlardır.

Dışsal Fonksiyon: Değerlendirmesi modelden bağımsız olarak dışssal etmenlerle değişen fonksiyonlardır. Modeldeki girişlerin tanımlanması ve rastsal değişimler dışsal fonksiyonlar sayesinde yapılır.

Paylaşımlı Fonksiyon: Model tarafından veya dışsal etmenlerle değerlendirilmesi yapılan fonksiyonlardır.

3.1.1.2 Geçişler ve Geçiş Kuralları

Geçişler bir durumdan diğer duruma geçerken dinamik fonksiyonların nasıl değişeceklerini tanımlar. Bu işlemler ASM de geçiş kuralları ile koşullandırılabilir. Bahsedildiği üzere statik fonksiyonların değişmesi söz konusu değildir. Ayrıca dışsal fonksiyonlar durumlar arası geçişlerde modelden bağımsız olarak yani deterministik olmayan bir şekilde değişirler.

Tanım 3.4 (*Güncelleme*): Konum değer ikilisi (*Kon,v*) , örneğin $(f,(v_1,...,v_n),v)$ bir güncelleme belirtir ve en temel durum değişimidir.

Tanım 3.5 (Güncelleme Kümesi) : Durumda gerçekleştirilebilecek güncellemeleri içeren kümeye *güncelleme kümesi* denir. Güncellemeler geçiş kuralları koşullarına göre olmaktadır. ASM de bir güncelleme kümesinin gerçekleşmesi için kümenin tutarlı olması gerekmektedir.

Tanım 3.6 (Tutarlı Güncelleme) : Bir *updSet* güncelleme kümesinin tutarlı olması için aşağıdaki koşulun sğlanması gerekir.

$$(((f,(a_1,...a_n)),b) \in updSet \wedge ((f,(a_1,...a_n)),c) \in updSet) \rightarrow b = c$$

Kısaca değinmek gerekirse bu şart eğer bir güncelleme kümesi tutarlı ise bir konum iki farklı değere güncellenemez anlamına gelmektedir.

Tanım 3.7 (Yürütme): Bir *yürütme* modelde sonlu veya sonsuz $s_0, s_1, ..., s_n, ...$ sekanslarından oluşmaktadır. Başlangıç durumu s_0 olmak üzere her bir s_{t+1} durumu s_t durumundaki geçiş kurallarının *eşzamanlı* olarak gerçekleştirilmesi ile belirlenir.

Geçiş Kuralları

Geçiş kuralları en basit şekilde *cond* koşulunun sağlanmasıyla gerçekleştirilen güncellemeler olarak tanımlanır.

If *cond* **then** *upd*

Burada *cond* bütün değişkenleri tanımlanmış bir formülü ifade etmekte ve *upd* bir fonksiyon güncellemelerinin kümesini belirtmekte olup elemanları $f(t_1,...,t_n) := t$ formundadır. *f* fonksiyonu *n* argümanlı bir fonksiyon ve $t_1,...,t_n,t$ birinci dereceden terimlerdir. *i>0* olmak üzere s_i durumunda bu geçiş kuralının gerçekleştirilmesi için öncelikle bütün $t_1,...,t_n,t$ terimleri incelenir, sonrasında ise $(f,(v_1,...,v_n))$ konumunun değeri güncellenerek *v* olur ve s_{i+1} durumunda değerlendirilmesi yapılır.

ASM de geçiş kuralları bazı sınıflara ayrılmıştır.

Atlama Kuralı: Herhangi bir fonksiyona etki yapmadan o adımın atlanmasını sağlar.

skip

Güncelleme Kuralı: $(f,(v_1,...,v_n))$ konumunu *v* değerine günceller. Burada $v_1...v_n$ ve *v* , $t_1,...,t_n$ ve *t* terimlerinin o andaki durumdaki değerlendirilmesi olarak tanımlanır.

$$f(t_1,...|,t_n) := t$$

Blok Kuralı: R_1 ve R_2 kurallarının eş zamanlı ve paralel olarak gerçekleştirilmesini sağlar.

$$R_1 \text{ par } R_2$$

veya daha genel olarak $n \geq 2$ olmak üzere,

par

R_1

...

R_n

endpar

şeklinde ifade edilebilir.

Şartlandırma Kuralı: Eğer lojik olarak tanımlanmış *cond* şartı *doğru* ise R_t kuralı uygulanır, eğer *doğru* değilse R_e kuralı geçerlidir.

If *cond* **then**

R_t

else

R_e

endif

"Hepsi için" Kuralı: Eğer herhangi bir $v \in D$ için lojik *cond* şartı *doğru* ise, şartı sağlayan *v* değerleri için *R* geçiş kuralı paralel olarak gerçeklenir.

forall *v* **in** *D* **with** *cond* **do**

R

Seçim Kuralı: Rastgele *cond* koşulunu sağlayan bir *v* değeri seçilir ve *R* geçiş kuralı gerçeklenir. Eğer *cond* koşulunu sağlayan herhangi bir *v* değeri bulunamazsa herhangi bir işlem yapılmaz.

Choose *v* **in** *D* **with** *cond* **do**

R

Ekleme Kuralı: Yedek bir kümeden, örneğin *X* süperuzayının bir alt kümesinden, bir *e* elemanını *D* tanım kümesinde ekleyerek *R* kuralının gerçeklenmesini sağlar.

Extend *D* **with** *e*

R

Çağırma Kuralı: $t_1,...,t_n$ parametrelerini kullanarak *R* kuralının çağrılmasını sağlar.

$$R(t_1,...,t_n)$$

Kelime hazinesi, durum ve geçiş kurallarını açıkladıktan sonra ASM modelini tanımlamaya geçilebilir.

Tanım 3.8 (Kural Tanımı): Bir $x_1,...,x_n$ değişkenlerini içeren *R* geçiş kuralı için *n* argümanlı bir r kıuralı aşağıdaki gibi tanımlanır,

$$r(x_1,...,x_n) = R$$

$r(t_1,...,t_n)$ olarak verilmiş bir çağırma kuralında her *R*'deki her x_i değişkeni bu kuraldaki t_i ile değiştirilir.

Tanım 3.9 (ASM) Bir ASM aşağıda verilen elemanlardan oluşur,

- Σ kelime hazinesi ile tüm fonksiyon sınıflandırılmaları,
- S_0 başlangıç durumları kümesi,
- *RD* kural tanımlamaları kümesi,
- ASM nin başlangıç noktasını oluşturan ve direkt veya dolaylı olarak *RD* kümesindeki kurallar ile oluşturulan küme.

3.2 Gereksinimlerin Modellenmesi

Sistemin modeli oluşturulduktan sonra sistemin sağlaması gereken gereksinimler ve sistemde oluşabilecek hataların kontrol edilebilmesi için bu önermeler zamansal lojik(temporal logic)[20] formülleriyle ifade edilmelidir. Zamansal lojik, lojik önermelere zamansal özellik kazandırır. Örneğin, aynı konveyör hattında ilerleyen farklı özelliklere sahip ürünlerin uygun kanallara yönlendirildiği, her ürün için farklı pistonun kullanıldığı bir proses düşünülsün. Her ürünün belli bir sırada geldiği kabul edilirse, her pistonun da belli bir sırada aktif olacağı aşikardır. Bu durumda pistonların sırasını belirtmek için zamansal ifadelere ihtiyaç duyulur. Örneğin "Piston1 pasif olduktan *sonra* Piston2 aktif olur" önermesindeki *sonra* ifadesi zamansal bir ifadedir ve önermeden çıkarılması durumunda önerme istenileni karşılamamaktadır. Başka bir örnek olarak "Piston1 aktifse *er ya da geç* pasif olacaktır." önermesi verilebilir. Burada da sırayla çalışan bir piston sisteminde diğer ürünün uygun kanala yönlendirilmesi için mutlaka pistonun aktif olduktan sonra pasif olması gerektiği vurgulanmaktadır. Göründüğü üzere zamansal ifadeler olmazsa istenilen anlam verilememektedir.

Gereksinimler ile sistemin sağlaması gereken veya sistemde oluşabilecek hata durumları belirtilmektedir. Yukarıda belirtildiği gibi "Piston1 pasif olduktan *sonra* Piston2 aktif olur" önermesi bir gereksinimdir. Bu önermenin lojik olarak tersi de hata durumunu verecektir.

NuSMV model kontrolü yazılımı LTL(Linear Temporal Logic) ve CTL(Computational Tree Logic) notasyonlarında tanımlanmış zamansal lojik önermelerin(gereksinimlerin) sağlanıp sağlanmadığını kontrol edebilmektedir. LTL, durumların lineer olarak incelenmesini sağlar. Bir durumun sadece bir tane ardıl durumu vardır. Ardıl durum, o andaki durumdan sonra gelebilecek olan durumlar olarak ifade edilebilir. CTL formulizayonunda ise bir durumun bir kaç tane ardıl durumu olabilir. Durumlar ve olası ardıl durumlar bir ağaç gibi dallanmış halde bulunurlar. Dikkat edilmesi gereken husus, bazı önermelerin hem CTL hem de LTL ile ifade edilemeyişidir. Bu durumda hangi zamansal lojik ifade biçiminin kullanılması gerektiği tamamen ne ifade edilmek istenildiğine bağlıdır. NuSMV daha önceden de belirtildiği gibi her iki formülasyon için de model kontrolü olanağı sağlamaktadır. Bu tek bir model için her iki formülizasyonu da kullanarak model kontrolünü olanaklı kılar.

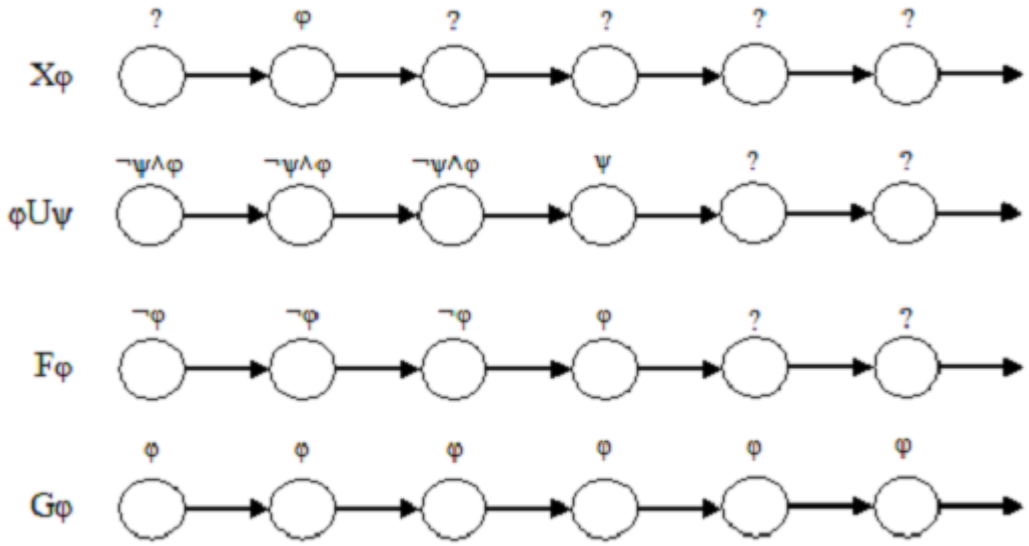
Gereksinimlerin modellenmesinin daha iyi anlaşılması açısından LTL ve CTL formulizasyonlarının anlamsal öğelerinden kısaca bahsetmek yararlı olacaktır. LTL daha önce de bahsedildiği gibi her zaman birimi için, ki bahsedilen zaman birimi $s_i \in S$, $i = 1,2,3...n$ olmak üzere s_i durumunu temsil etmekte olup, sadece bir ardıl durumun incelendiği formulizasyondur. LTL formülleri, ϕ ve ψ atomik önermeler olmak üzere, **¬, ^, ˅**, $\rightarrow, \leftrightarrow$ sırasıyla, "değil", "ve", "veya", "ise", "ancak ve ancak" lojik operatörlerini ve Xϕ (ϕ'den sonra), ϕUψ (ψ'ye kadar ϕ atomik önermesi geçerlidir.), Gϕ (ϕ her zaman geçerlidir), Fϕ (ϕ er ya da geç geçerlidir) zamansal ifadelerini içermektedir.

"G" ve "F" zamansal modaliteleri, "U" opertörü ile türetilirler. Eşlenik anlamsal ifadeler aşağıda gösterilmiştir[21].

Fϕ ≡ doğru **U** ϕ , **G**ϕ ≡ ¬ **F** ¬ϕ

LTL formüllerinin diyagram üzerindeki gösterimi Şekil 3.2 deki gibidir. Diyagramda durumlar çemberler ile gösterilmiş ve durumlarda sağlanan önermeler, çemberlerin

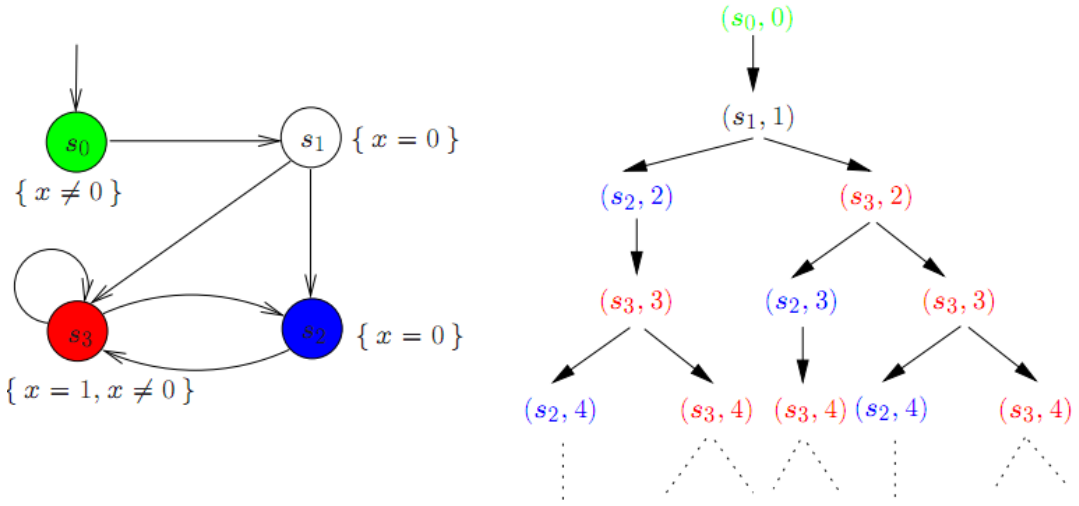
üzerinde bulunmaktadır. "?" sembolü ile bu durumdaki önermenin formülün sağlanmasıyla ilgisinin olmadığı gösterilmektedir.



Şekil 3. 2 LTL formüllerinin gösterimi

Zamansal modaliteler birlikte kullanılarak yeni zamansal modaliteler elde edilir. GFϕ (sonsuz olarak sık sık ϕ geçerlidir) zamansal modalitesi herhangi bir j anında $i \geq j$ olacak şekilde ϕ önemesinin sağlandığı bir duruma uğranmıştır anlamına gelirken, FGϕ (er yada geç sonsuza kadar ϕ geçerlidir) zamansal modalitesi herhangi bir j anından sonra sadece ϕ önemesinin sağlandığı durumlara uğranmıştır anlamına gelmektedir.

CTL formülizasyonu durumlar ve olası durumların bir ağaç gibi dallanarak incelenmesini sağlar. Önceden bahsedildiği gibi bir durumun olası bir kaç ardıl durumu bulunmaktadır. Farklı ardıl durumlar dallanmasıyla bütün olası yollar elde edilir. Yol başlangıç durumundan son durumlara kadar olan olası durumların dizilimidir. Şekil 3.3 de bir CTL örneği gösterilmiştir [22].

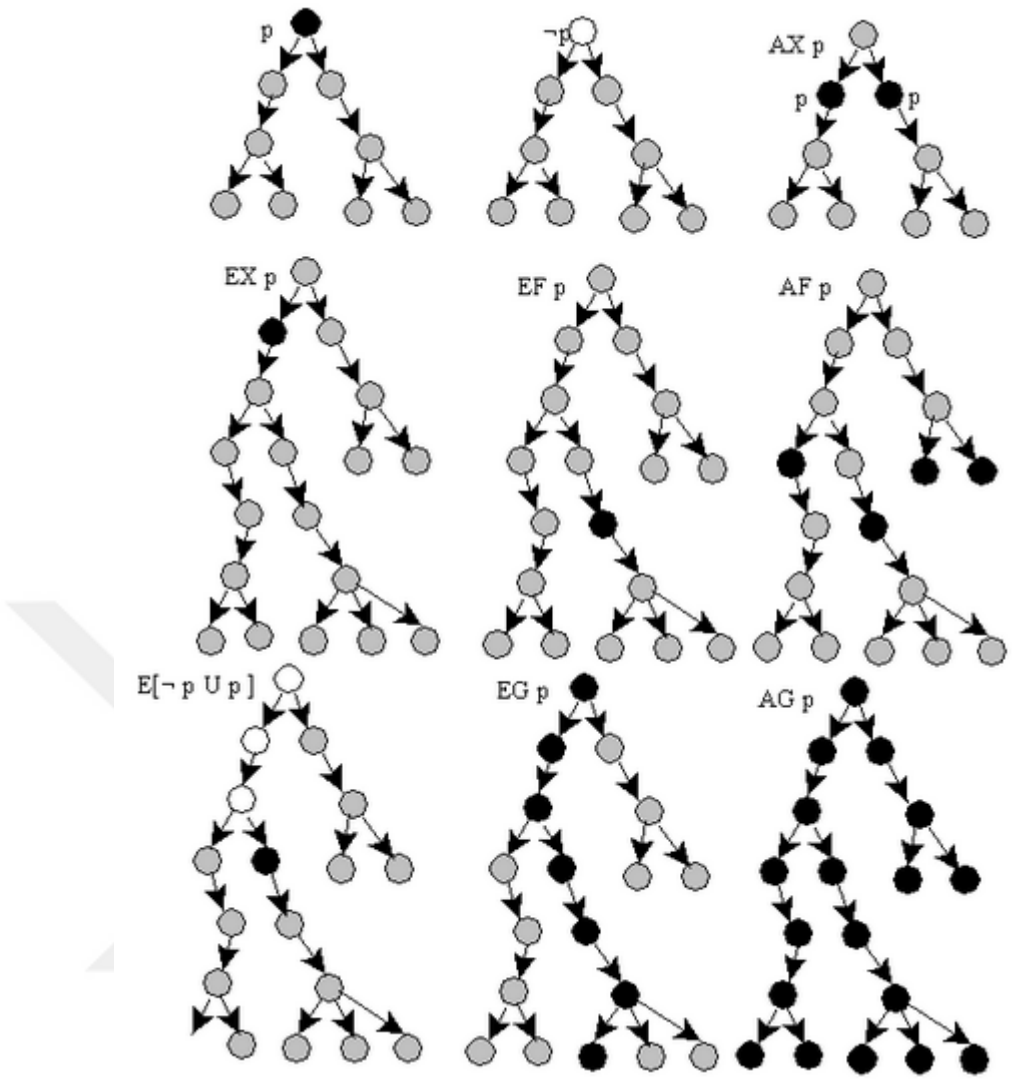


Şekil 3. 3 CTL örneği

Şekil 3.2 de gösterilen CTL örneğinde $s_0 \to s_1 \to s_2 \to s_3 \to s_3$ olası bir yol olmak üzere $s_0 \to s_1 \to s_3 \to s_2 \to s_3$ farklı bir olası yoldur. CTL formülünde, LTL formüllerinde kullanılan X,U, G, F operatörleri aynı anlamlara sahiptir ve durum niceleyicileri adını alır. Bunların yanında yol niceleyici operatörler mevcuttur. Bunlar "A"(Bütün olası yollar için) ve "E"(Bazı olası yollar için) operatörleridir. Örneğin atomik önermeler uzayı $AP = \{x < 1, x = 1, x > 1\}$ olsun. Bütün olası yollar için yol boyunca $x = 1$ atomik önermesi sağlanmaktadır ifadesinin CTL formülü ile ifade edilişi aşağıdaki gibidir.

$$\text{AG}(x = 1)$$

Konunun daha net anlaşılmasını sağlamak adına Şekil 3.4 te CTL formüllerinin sağlandığı durumlar diyagram üzerinde gösterilmiştir[23]. Siyah daire ile gösterilen durumlar p önermesinin geçerli olduğu durumları ifade etmektedir.



Şekil 3. 4 CTL formüllerine örnekler

3.3 Karşı Örnek Analizi

Karşı örnek, model kontrolü gerçekleştirildikten sonra kontrol edilen özelliklerin karşılanıp karşılanmadığının anlaşılmasını sağlar. Karşı örnek eğer gereksinimler karşılanmışsa mantıksal doğru şeklindedir. Eğer gereksinimler karşılanmamışsa, gereksinimlerin sağlanmadığı herhangi bir durum ve bu duruma ulaşmak için izlenen örnek yol gösterilir. Yol daha önceden de bahsedildiği gibi durumlar ve ardıl durumlarının oluşturduğu bir yapıdır ve karşı örnek için belirlenen yol için son durum gereksinimin karşılanmadığı durumdur.

BÖLÜM 4

NuSMV MODEL KONTROLÜ YAZILIMI

NuSMV model kontrolü için geliştirilmiş sembolik model kontrolü yazılımıdır. ITC-IRST ve UniTN tarafından CMU ve UniGE işbirliği ile geliştirilmiştir. NuSMV yazılımının yaratıcıları aşağıdaki özellikleri sağlayabilmesini hedeflediklerini belirtirler :

- Kararlı, geliştirilebilir ve özelleştirilebilir,
- Teknoloji transfer projelerinde uygulanabilir,
- Farklı disiplinlerde araştırma aracı olarak hizmet edebilmesi.

NuSMV küçük durum makinelerinin basitçe modellenebilmesini sağlarken, karmaşık modeller için de yapısal kolaylıklar sağlamaktadır. Durum makineleri daha önce de bahsedildiği gibi genel olarak durumlardan ve geçişlerden oluşur. NuSMV de durumlar kısaca durum değişkenlerinin kümesidir. Bu değişkenler

- Boole Değeri : {Doğru , Yanlış},
- Sayı aralığı : 1..9
- Sembolik küme : {İleride, Ortada, Geride} veya aynı tip değişkenlerin oluşturdukları “Diziler” olarak tanımlanabilirler.

Durum değişkenleri “VAR” anahtar kelimesi ile tanımlanırlar:

```
VAR
b0 : boolean;
i :  0..9;
st : {blue,red,yellow};
```

Durum makinelerinin bir durumdan başka bir duruma nasıl ve hangi koşullarda geçileceğini geçişler belirlemektedir. Geçişler tanımlanmadan önce ilk durum belirlenmelidir. Eğer bir durum için ilk durum belirlenmemişse bu durum deterministik olmayan bir şekilde, durum uzayında kısıtlı kalmak suretiyle rastgele olarak, değerlenecektir. Durumların ilk durumu “init()” fonksiyonu ile atanmaktadır. Geçişler ise “next()” fonksiyonu ile belirlenmektedir. Bu atamalar için “ASSIGN” anahtar kelimesi kullanılır.

```
VAR
b0 : boolean;
i :  0..9;
st  : {blue,red,yellow};

ASSIGN
init(b0) := TRUE;
next(i)   := (i+1) mod 10 ;
```

Yukarıda gösterilen örnekte “b0” durum değişkeninin ilk değeri “TRUE” olarak atanmış ve “i” durum değişkeninin bir sonraki değeri “i” durum değişkeninin şimdiki değerinin 1 arttırılması ve modunun alınması olarak belirlenmiştir. “b0” ve “st” durum değişkenlerinin bir sonraki durumdaki değerleri belirlenmediğinden bunların gelecekteki değerleri rastgele olarak belirlenecektir. Durum değişkenlerinin gelecek durumdaki değerlerinin belirlenmemesi deterministik olmayan durum makinelerinin kontrolünü olanaklı kılmaktadır.

Durum makinelerinin modellenmesinde bazı koşullar veya seçimler de kullanılabilmektedir. Koşullandırma için “IF THEN ELSE” yapısı (Cond ? exp1 : exp2) olarak tanımlanmıştır. Burada “Cond” şart, “exp1” şartın doğru olma koşulu ile gerçekleştirilecek işlem ve “exp2” şartın yanlış olması durumda gerçekleştirilecek olan işlemdir. Bunun yanında “Case.. Esac” seçim işlemi de kullanılabilmektedir.

*case expr :: **case***

expr a1 : expr b1 ;

. . .

expr an : expr bn ;

TRUE : default ;

esac

“Case..Esac” seçim operatöründe ilk sağlanan koşul için işlemler gerçekleştirilir diğer işlemler o durum için gerçekleştirilmez.

Karmaşık modellerin basite indirgenebilmesi, modellerin aynı işlevleri gören daha basit modellere ayrılması ile mümkündür. NuSMV bunun için “MODULE” kavramını bulundurmaktadır. “MODULE” olarak tanımlanan modeller farklı girişler tanımlanarak birbirlerinden ayrılırlar ve hepsi aynı yapıya sahip olmasına rağmen birbirlerinden

bağımsız olarak durum değişkenlerinin değişmesi sağlanır. Bütün modüller “main” ana modülü altında işlem görür. Aşağıda bir modül yapısı gösterilmiştir:

```
MODULE counter(inc)
VAR digit : 0..9;
ASSIGN
init(digit) := 0;
next(digit) := inc ? (digit + 1) mod 10 : digit;
DEFINE
top := digit = 9;


MODULE main
VAR
c0 : counter(TRUE);
c1 : counter(c0.top);
sum : 0..99;
 ASSIGN
sum := c0.digit + 10 * c1.digit;
```

NuSMV model kontrol yazılımı ile spefikasyonları belirlemek için iki zamansal lojik yapısı kullanılır. Bunlar LTL ve CTL zamansal lojik dilinde yazılmış olan spesifikasyonlardır. LTL ve CTL zamansal lojiği ve yazım kuralları hakkında daha ayrıntılı bilgi 3. bölümde anlatılmıştır.

NuSMV ile model kontrolü yapmak için “NuSMV Interactive” uygulaması kullanılmaktadır. Kullanımı hakkında daha ayrıntılı bilgi “NuSMV Manual”[28] dökümanında anlatılmıştır.

BÖLÜM 5

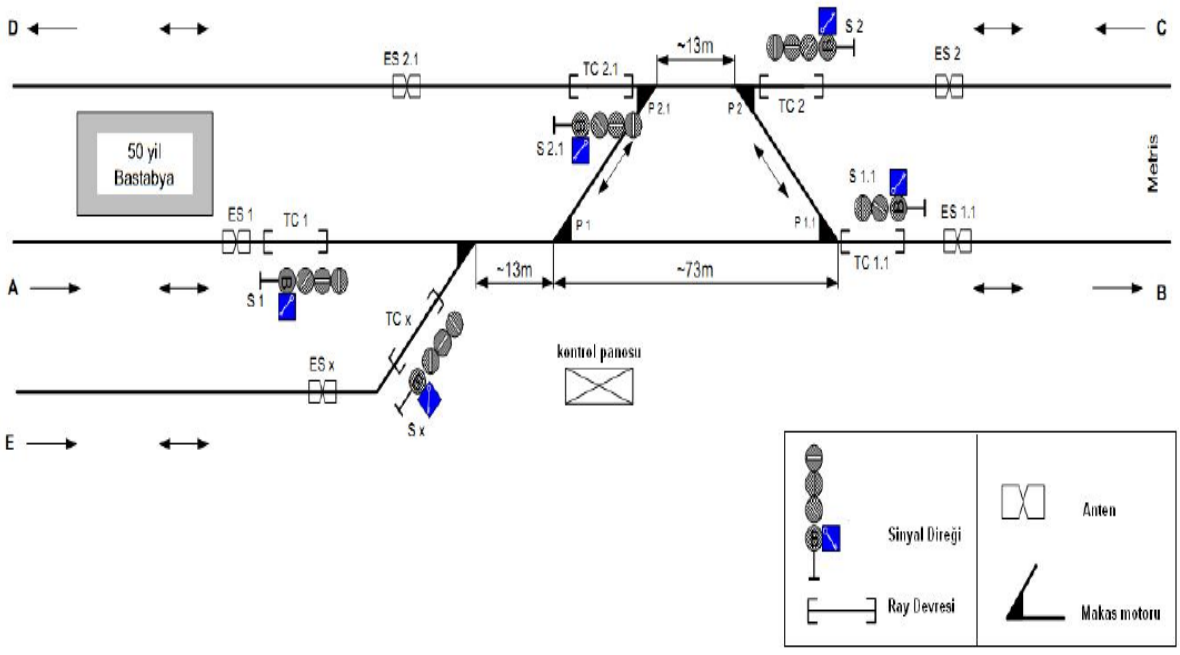
KONVANSİYONEL RAYLI SİSTEMLERİN ANKLAŞMAN TABLOSUNUN MODEL KONTROLÜ

Anklaşman sistemleri, trenlerin rotalarının güvenli bir şekilde ayarlanmasını sağlayan sistemlerdir. Tren rotasının açılması için bazı özel koşulların sağlanması gerekir. Ray devreleri, makas konumları, sinyal durumları ve diğer rotaların kullanım durumu anklaşman sistemlerinde rotaların açılma şartlarında genel olarak kullanılan öğelerdir. Anklaşman sistemlerinin yapıları , ülkelere, kurumlara, topolojiye ve farklı kullanım amaçlarına göre değişiklik gösterir. Bu çalışmada yapılmak istenen her anklaşman sisteminde genel olarak bulunan öğeler kullanılarak oluşturulan anklaşman sisteminin modeli oluşturularak, çekirdek bir model üzerinden model kontrolü gerçekleştirmektir.

Hızlı trenlerin anklaşman sistemleri de genel olarak konvansiyonel raylı ulaşım sistemlerinin anklaşman sistemleri gibi ray devreleri, sinyaller, makasların pozisyonları, çakışan rotalar, kanat koruması gibi öğelerin yanında trenlerin hız bilgisini de içerir. Model kontrolü için öncelikle konvansiyel raylı ulaşım sistemlerinin anklaşman sistemlerini inceleyerek başlamak ve öncelikle bu alandaki sorunlar çözüldükten sonra hızlı trenler için gerekli değişikliklerin eklenmesi uygun bulunmuştur.

Anklaşman sistemlerinin modellenmesinde üç ayrı unsur dikkate alınmıştır. Bunlar topoloji, anklaşan tablosu ve tren hareketidir. Topoloji modeli ray bölgelerinin, makasların ve sinyalerin konumlarını ve durumlarını içermektedir. Anklaşman tablosunun modeli rotaların açılması için gereken koşulların ve açılacak rota için yapılması gereken durum değişikliklerinden oluşmaktadır. Tren hareketinin modeli trenin topolojiye ve sinyallere göre nasıl hareket edeceğini belirlemektedir.

Yapılacak model kontrolü uygulaması için "50. Yıl Bastabya" istasyonunun topolojisi kullanılmıştır. Şekil 5.1 de görüldüğü üzere topoloji ray bölgeleri, makaslar, sinyaller ve antenlerden oluşmaktadır. Antenler trenlerin rota isteğini iletmektedir. Sinyaller düz çizgi, dik çizgi ve yan çizgi olmak üzere üç çeşittir. Düz çüzgi trenin rotasına düz bir şekilde devam edebileceğini, dik çizgi trenlerin geçişine izin verilmediğini ve yan çizgi ise makas üzerinden rotanın çizildiğini işaret etmektedir. Makasların pozisyonları açılan rotaya göre değişmektedir. Ray bölgeleri ise üzerinde tren olup olmadığının anlaşılmasını sağlamaktadır. Görüldüğü üzere ray devreleri sinyallerden sonra yer almaktadır. Yani bu ray bölgesi geçildikten sonra trenin konumu hakkındaki bir sonraki bilgi ancak rota tamamlanırken alınmaktadır. Orta nokta olarak adlandırılabilecek olan ray devrelerinin arasında bulunan kısımda bir tehlike oluşmaması için anklaşman sistemi çok dikkatli hazırlanmalıdır. Güvenliği garanti edilemeyen bir anklaşman sisteminin tasarlanması, herhangi bir koşulun unutulması, sonucu kazaların olabileceği model kontrolü ile yapılan çalışma sonucu gözler önüne serilmiştir.



Şekil 5. 1 "50. Yıl Bastabya" Topolojisi

Modellemeye başlarken öncelikle sistem statik yapısı yani topolojisi ele alınmıştır. ASM modelde bunlar statik fonksiyon olarak düzenlenmiştir. Her çeşit elemana ait bir alt kelime hazinesi uzayı belirtilmiştir.

Track: {Trackx, Track1, Track11, Track2, Track21}
-TrackName: {TCx, TC1, TC11, TC2, TC21}
Point:{Pointx, Point1, Point2}
-PointName:{Px, P1, P2}
-PointPosition:{Normal, Inverse}
Antenna:{Antennax, Antenna1, Antenna11, Antenna2, Antenna21}
-AntennaName:{ESx, ES1, ES11, ES2, ES21}
Signal:{Signalx, Signal1, Signal11, Signal2, Signal21}
-SignalName:{Sx, S1, S11, S2, S21}
-SignalColour:{Red,Green,Yellow}
Route:{Route1,.., Route11}

Sistemde herhangi bir arıza durumu olmadığı kabul edildiğinden P1 ile P21 makas motorları P1, P2 ile P11 makas motorları P2 olarak adlandırılarak fazla değişken tanımlamaktan kaçınılmıştır. Kabul edilen şart karşılıklı makasların pozisyonlarının aynı anda ters ya da aynı anda düz oldukları şeklindedir.

Ray bölgelerinin isimleri değişmeyeceğinden bunlar statik fonksiyonlar olarak tanımlanmıştır. Ray bölgelerinin meşguliyeti tren ray bölgesine gelip gittiğinde değişeceğinden dinamik fonksiyon olarak ele alınmıştır. Sinyallerin durumları açılan rotalara ve ray bölgelerinin meşguliyetine göre değişeceğinden yine statik fonksiyonlar olarak ifade edilmiştir.

Tren hareketi modeli fazla karmaşık tutmamak adına her seferinde tek bir ray bölgesi geçebilecek şekilde modellenmiştir. Trenin uzunluğu veya hızı bu modelde ihmal edilmiştir ve en fazla iki ray bölgesini kaplayabilecek şekilde modeli oluşturulmuştur. Yani trenin önü ve arkası farklı ray bölgelerinde bulunabilmektedir. Trenler sinyallere ve makasların durumlarına göre hareket etmektedirler. Bu modelde karşı karşıya hareket eden iki tren kullanılmıştır fakat esnek bir model oluşturulduğundan daha fazla tren eklenebilir.

static function *TrackID : Track --> TrackName*

static function *PointID : Point --> PointName*

static function *SigID : Signal --> SignalName*

dynamic function *Occ : Track --> Boolean*

dynamic function *SigColour: Signal --> SignalColour*

dynamic function *RouteLock: Route--> Boolean*

dynamic function *RouteFree: Route--> Boolean*

static function *RouteEnd : Route --> AntennaName*

dynamic function *PointPos : Point--> PointPosition*

dynamic function *PointLock: Point--> Boolean*

Trenlerin başlangıç konumu, rastgele olacak şekilde ayarlanmıştır. Ayrıca trenlerin rota istekleri de rastgele seçilerek, olası tüm durumların taranması amaçlanmıştır. Trenlerin isimleri ve yönleri statik fonksiyonlar olarak tanımlanıp trenin konumu dinamik fonksiyon olarak belirlenmiştir. Bahsedilen deterministik olmayan davranışlar için dışsal fonksiyonlar kullanılmıştır.

Train : {Train 1, Train 2,...,Train **n**}

-TrainName : {Train_1, Train_2,...,Train_n}

-Direction : {Left, Right}

static function TrainID : Train --> TrainName

static function TrainDirection : Train --> Direction

dynamic function TrainFrontSituationID : Train --> TrackName ∪ PointName ∪ AntennaName

dynamic function TrainBackSituationID : Train --> TrackName ∪ PointName ∪ AntennaName

external function TrainRequestID : Train --> AntennaName

external function Train_InitSituation : Train --> AntennaName

Fonksiyonların tanımlanması ile modelin tabanı oluşturulmuştur. Modelin evrilmesini sağlayacak olan yapı yani durumlar arası geçişler daha önceden de bahsedildiği geçiş kuralları ile sağlanmaktadır. Anklaşman sistemindeki şartlara göre sinyalizasyon yapılması, makasların kontrolü ve tren hareketinin modellenmesi geçiş kurallarının tanımlanmsı ile mümkün kılınmıştır. Çizelge 5.1 de incelemek için oluşturulan anklaşman tablosu örneği gösterilmektedir. Anklaşman tablosu tanımlamalar ile karışıklık oluşturmaması açısından ingilizce olarak hazırlanmıştır.

Çizelge 5.1 Tasarlanan Anklaşman Tablosu

Route	**Rail Circuits**	**Points**	**Signals**	**Counter Routes**
R1: ESx-->TC11	TCx, TC11	Px : Inverse P1 : Normal	Sx: Y S1: R S21: {G,R} S2:{G,R}	All / {R1,R6,R10}
R2: ESx --> TC2	TCx, TC2	Px : Inverse P1: Inverse P2: Normal	Sx: Y S1: R S21: R	All
R3:ES1 --> TC11	TC1, TC11	Px : Normal P1 : Normal P2 : Normal	Sx : R S1 : G S21 : {G,R}	All / {R3,R6,R10}
R4: ES1 --> TC2	TC1, TC2	Px : Normal P1 : Inverse P2 : Normal	Sx : R S1 : Y S21 : R	All
R5: ES21 --> TC11	TC21, TC11	P1 : Normal P2 : Inverse	S21 : Y S2 : R	All
R6: ES21 --> TC2	TC21, TC2	P1 : Normal P2 : Normal	S21 : G	All / {R6,R1,R3}

Çizelge 5.1 Tasarlanan Anklaşman Tablosu(devamı)

R7: ES11 -->TC21	TC11, TC21	P1 : Normal P2 : Inverse	S11 : Y S2 : R	All
R8 : ES2 --> TCx	TC2, TCx	Px : Inverse P1 : Inverse P2 : Normal	S2 : Y S21 : R S1 : R S11 : R	All
R9: ES2 --> TC1	TC2, TC1	Px : Normal P1 : Normal P2 : Normal	S2 : Y S21: R S1 : R Sx : R	All
R10: ES2 --> TC21	TC2, TC21	P1 : Normal P2 : Normal	S2 : G	All / {R10,R1,R3}
R11 : ES11 --> TCx	TC11, TCx	Px : Inverse P1 : Normal P2 : Normal	S11: G	All /{R11,R6,R10}

Fonksiyonların tanımlanması ile modelin tabanı oluşturulmuştur. Modelin evrilmesini sağlayacak olan yapı yani durumlar arası geçişler daha önceden de bahsedildiği geçiş kuralları ile sağlanmaktadır. Anklaşman sistemindeki şartlara göre sinyalizasyon yapılması, makasların kontrolü ve tren hareketinin modellenmesi geçiş kurallarının tanımlanmsı ile mümkün kılınmıştır. Çizelge 5.1 de incelenmek üzere oluşturulan anklaşman tablosu örneği gösterilmektedir. Anklaşman tablosu tanımlamalar ile karışıklık oluşturmaması açısından ingilizce olarak hazırlanmıştır.

Başlangıç durumunda rotalar ve makaslar kilitli değildir. Sinyallerin hepsi geçişe izin vermeyecek şekilde, yatay çizgi(kırmızı), olarak tanımlanmıştır. Daha önceden bahsedildiği gibi trenlerin rota istekleri ve başlangıç konumları ise rastgele belirlenmektedir. Başlangıç durumunda statik ve dinamik fonksiyonların ilk değerleri verilir. Örneğin,

TrackID(Track1):= TC1

Sistemin elemanları anklaşman tablosuna göre değer alır. Örneğin bir tren R1 rotasını talep ettiyse ve bu rotanın açılması için gereken şartlar sağlandıysa, yani ray bölgeleri boş ve karşıt rotalardan herhangi biri kilitlenmemiş durumda ise , o rota kilitlenir. Rota kilitlendikten sonra makaslar ayarlanır ve makaslar kilitlenir. Sonra da trene gerekli sinyal verilir.

R1 rotasının kilitlenmesi için oluşturulan geçiş kuralı aşağıdaki gibir. Diğer rotalar için de benzer geçiş kuralları hazırlanmıştır. Burada belirtilen "*guardRF*" ile belirtilen koşul karşıt rotaların kilitlenmemiş olması (*routelock(R)=FALSE*) ve ilgili ray bölgelerinin boş olması(*Occ(Trackx)=FALSE* **and** *Occ(Track11)=FALSE*) olarak tanımlanmıştır.

if *guardRF* **then**
routefree(Route1):=TRUE
else
routefree(Route1): FALSE

forall *R* **in** *Route* **with** (*routefree(R)* **and** *(TrainRequest(Train1) = RouteEnd(R)* **or** *TrainRequest(Train2) = RouteEnd(R))*
routelock(R)

Trenin hareketi sinyallere, makas durumlarına ve trenin hareket yönüne göre belirlenmektedir. Basit olarak tren her seferinde bir ray bloğu atlamaktadır. Eğer önündeki bir ray bloğu ise ve herhangi bir sinyalizasyon sınırılaması yoksa tren koşulsuz olarak önündeki ray bloğuna geçer. Eğer hareket yönünde bir makasla karşılanmışsa makasun pozisyon durumuna göre trenin gidebileceği farklı yollar bulunmaktadır.

ASM notasyonu ile NuSMV notasyonu arasında belirgin benzerlikler vardır. ASM modelinde tanımlanan fonksiyon NuSMV de *VAR* veya *FROZENVAR* olarak tanımlanan değişkenlerle ifade edilebilmektedir. Örneğin ASM modelinde tanımlanan statik fonksiyon TrackID ve dinamik fonksiyon Occ, NuSMV de aşağıdaki gibi ifade edilir.

static function *TrackID : Track --> TrackName*

dynamic function *Occ : Track --> Boolean*

ASM

FROZEN VAR TrackID : *{TCx, TC1, TC11, TC2, TC21}*

VAR Occ : Boolean

NuSMV

Fonksiyonların tanımlandığı tanım kümeleri ve değer kümeleri NuSMV de *MODULE* olarak tanımlanan üst nesnelerin alt nesleri olarak tanımlanabilmektedir. Üst nesneler bir çok elemanın tanımlanabileceği yapılar olarak görev alır.

Point:{Pointx, Point1, Point2}
-PointName:{Px, P1, P2}
-PointPosition:{Normal, Inverse}
static function *PointID : Point --> PointName*
PointID(Pointx):=Px

ASM

MODULE *Point*
FROZENVAR *PointID: {Px,P1,P2};*
VAR *Position : {Normal,Inverse};*
VAR *Locked: boolean;*

MODULE *main*
VAR *Pointx: Point*
ASSIGN
init*(Pointx.PointID) := Px*

NuSMV

Geçiş kuralları "next" ifadesi ile belirtilmekte başlangıç durumları ise "init" ifadesi ile belirlenmektedir. ASM modelindeki

if *cond* **then** *upd*

yapısı NuSMV modelinde,

next(*Data)* := **case** *cond* **:** *upd*
TRUE **:** *Data*
esac

yapısı ile ifade edilebilmektedir.

ASM ile NuSMV arasında yapılabilen bu dönüşümler sayesinde tasarlanan ASM notasyonuyla ifade edilen anklaşman sistemi modeli, basit bir işlemle NuSMV notasyonuyla ifade edilebilmiştir.

Model oluşturulduktan gereksinimlerin belirlenmesi gerekmektedir. Gereksinimler sistemin sağlaması gereken önermelerden ibarettir. Daha önce de bahsedildiği gibi NuSMV model kontrolü yazılımında sistem gereksinimleri LTL veya CTL olak üzere iki şekilde ifade edilebilmektedir. Bu çalışmada gereksinimler CTL notasyonuyla tanımlanmıştır.

Sistemde herhangi bir arıza olmadığı kabul edildiğinden anklaşman sisteminde yapılabilecek olan bir hatayı tespit etmek adına gereksinimler olarak sistemin sağlaması gereken iki önemli unsur kontrol edilmiştir. Bunlardan biri trenlerin çarpışması diğeri ise trenin raydan çıkması durumudur. Trenlerin çarpışmasını kontrol etmek için aşağıdaki CTL formülü kullanılmıştır.

```
SPEC NAME Collusion := AG !(Train1.SituationID=Train2.SituationID
                | Train1.NextSituationID=Train2.SituationID
                | Train1.SituationID=Train2.NextSituationID);
```

Formülden anlaşılacağı üzere eğer trenlerin veya gelecek durumdaki konumları aynı olursa bu çarpışma anlamına anlamına gelecektir. Formül bütün olası yollar ve yol üzerindeki herhangi bir durum için bu önermeyi sağlayan bir durumun olmadığının kontrol edilmesini sağlar. Kontrol edildiğinde tasarlanan anklaşman sistemi için çarpışma durumunun olmadığı gözlenmiştir. Fakat modelin doğru olup olmadığının kontrolü ve model kontrolünün doğru sonuçlar verdiğini görebilmek için sisteme bir hata eklenmiştir. Hata olarak anklaşman tablosunun hazırlanmasında "Rota 9" için karşı rotalardan "Rota1" in unutulduğu varsayılmıştır. Bu durumda karşı örnek Çizelge 5.2 deki gibidir.

Çizelge 5.2 de görüldüğü gibi Rota1 ve Rota9 karşıt rotalar olmasına rağmen anklaşman sisteminde tanımlanmadığından Rota 1 kilitli iken Rota 9 da kilitlenebilmiştir. Bu hata trenlerin çarpışması ile sonuçlanmıştır.

Çizelge 5.2 NuSMV karşı örneği

--specification AG !((Train1.SituationID = Train2.SituationID | Train1.NextSituationID = Train2.SituationID) | Train1.SituationID = Train2.NextSituationID)

is false

-- as demonstrated by the following execution sequence

Trace Description: CTL Counterexample

Trace Type: Counterexample

-> State: 1.1 <-

Trackx.TrackID = TCx

Track1.TrackID = TC1

Track11.TrackID = TC11

Track2.TrackID = TC2

Track21.TrackID = TC21

Antennax.TrackID = ESx

Antenna1.TrackID = ES1

Antenna11.TrackID = ES11

Antenna2.TrackID = ES2

Antenna21.TrackID = ES21

Pointx.PointID = Px

Point1.PointID = P1

Point2.PointID = P2

Signalx.SigID = Sx

Signal1.SigID = S1

Signal11.SigID = S11

Signal2.SigID = S2

-> State: 1.2 <-

Pointx.Position = Inverse

Pointx.Locked = TRUE

Point1.Locked = TRUE

Point2.Locked = TRUE

Route1.Locked = TRUE

Route1.Free = FALSE

Route2.Free = FALSE

Route8.Free = FALSE

InitTrack_1 = ES21

Request_1 = ES2

Request_2 = ES21

Signal21.SigID = S21

Train1.TrainID = Train_1

Train1.Direction = Right

Train2.TrainID = Train_2

Train2.Direction = Left

Trackx.Occ = FALSE

Track1.Occ = FALSE

Track11.Occ = FALSE

Track2.Occ = FALSE

Track21.Occ = FALSE

Antennax.Occ = TRUE

Antenna1.Occ = FALSE

Antenna11.Occ = FALSE

Antenna2.Occ = TRUE

Antenna21.Occ = FALSE

Pointx.Position = Normal

Pointx.Locked = FALSE

Point1.Position = Normal

Point1.Locked = FALSE

Point2.Position = Normal

Point2.Locked = FALSE

Signalx.Colour = Red

Signal1.Colour = Red

Signal11.Colour = Red

Signal2.Colour = Red

Signal21.Colour = Red

Train1.SituationID = ESx

Train1.NextSituationID= ESx

Train1.Request = ES11

-> State: 1.3 <-

Point1.Position = Inverse

Signalx.Colour = Yellow

Train1.NextSituationID = TCx

Route9.Locked = TRUE

Route9.Free = FALSE

Route10.Free = FALSE

Train2.SituationID = ES2

Train2.NextSituationID= ES2

Train2.Request = ES1

Route1.Locked = FALSE

Route1.Free = TRUE

Route2.Locked = FALSE

Route2.Free = TRUE

Route3.Locked = FALSE

Route3.Free = FALSE

Route4.Locked = FALSE

Route4.Free = FALSE

Route5.Locked = FALSE

Route5.Free = FALSE

Route6.Locked = FALSE

Route6.Free = FALSE

Route7.Locked = FALSE

Route7.Free = FALSE

Route8.Locked = FALSE

Route8.Free = TRUE

Route9.Locked = FALSE

Route9.Free = TRUE

Route10.Locked = FALSE

Route10.Free = TRUE

Route11.Locked = FALSE

Route11.Free = FALSE

InitTrack_1 = ESx

InitTrack_2 = ES2

Request_1 = ES11

Request_2 = ES1

Çizelge 5.2 NuSMV karşı örneği(devamı)

-> State: 1.4 <-	-> State: 1.5 <-	
Trackx.Occ = TRUE	Trackx.Occ = FALSE	
Antennax.Occ = FALSE	Track2.Occ = TRUE	
Signal2.Colour = Yellow	Antenna2.Occ = FALSE	
Train1.SituationID = TCx	Signalx.Colour = Red	
Train1.NextSituationID = TC2	Train1.SituationID = TC2	
Train2.NextSituationID = TC2	Train2.SituationID = TC2	
----------	----------	

Trenlerin raydan çıkması kontrol edilmesi gereken bir diğer unsuru oluşturmaktadır. Eğer trenlerden birisi için rota açılmış fakat makas pozisyonlarında bir yanlışlık yapılmışsa bu durumda tren ya istediği rotadan farklı bir rota izleyecektir, ya da raydan çıkma durumları söz konusu olacaktır. Bu durumların tespiti için aşağıdaki CTL formülleri tanımlanmıştır.

SPEC NAME Derailment1 := AG !(Train1.SituationID = TC1 & Train1.Direction = Right & Pointx.Position = Inverse);

SPEC NAME Derailment2 := AG !(Train1.SituationID = TCx & Train1.Direction = Right & Pointx.Position = Normal);

SPEC NAME Derailment3 := AG !(Train1.SituationID = TC21 & Train1.Direction = Right & Point1.Position = Inverse);

SPEC NAME Derailment4 := AG !(Train2.SituationID = TC2 & Train2.Direction = Left & Point2.Position = Inverse);

SPEC NAME Derailment5 := AG !(Train2.SituationID = TC11 & Train2.Direction = Left & Point2.Position = Normal & Point1.Position=Inverse);

SPEC NAME Derailment6 := AG !((Train1.SituationID = TC1 | Train1.SituationID = TCx) & Train1.Direction = Right & Point1.Position = Normal & Point2.Position = Inverse);

Derailment2 ile adlandırılmış olan gereksinimde trenin TC21 ray bölgesinde bulunduğunda ve trenin yönünün sağa doğru olması ile Px makasının yanlış olarak normal ayarlanıp ayarlanmadığı kontrol edilmektedir. Tasarlanan anklaşman sistemi bunu sağlamaktadır fakat yine Rota1 de yanlışlık yapıldığı düşünüldüğünde NuSMV tarafından aşağıdaki karşı örnek alınmıştır.

Çizelge 5.3 Raydan çıkma durumu için karşı örnek

--specification AG !((Train1.SituationID = TCx & Train1.Direction = Right) & Pointx.Position = Normal) is false -- as demonstrated by the following execution sequence Trace Description: CTL Counterexample Trace Type: Counterexample **-> State: 1.1 <-** Trackx.TrackID = TCx Track1.TrackID = TC1 Track11.TrackID = TC11 Track2.TrackID = TC2 Track21.TrackID = TC21 Antennax.TrackID = ESx Antenna1.TrackID = ES1 Antenna11.TrackID = ES11 Antenna2.TrackID = ES2 Antenna21.TrackID = ES21 Pointx.PointID = Px Point1.PointID = P1 Point2.PointID = P2 Signalx.SigID = Sx Signal1.SigID = S1 Signal11.SigID = S11 Signal2.SigID = S2 Signal21.SigID = S21 Train1.TrainID = Train_1 Train1.Direction = Right Train2.TrainID = Train_2	**-> State: 1.2 <-** Pointx.Locked = TRUE Point1.Locked = TRUE Point2.Locked = TRUE Route1.Locked = TRUE Route1.Free = FALSE Route2.Free = FALSE Route8.Free = FALSE Route9.Free = FALSE InitTrack_1 = ES21 Request_1 = ES2 Request_2 = ES21 Train2.Direction = Left Trackx.Occ = FALSE Track1.Occ = FALSE Track11.Occ = FALSE Track2.Occ = FALSE Track21.Occ = FALSE Antennax.Occ = TRUE Antenna1.Occ = FALSE Antenna11.Occ = FALSE Antenna2.Occ = TRUE Antenna21.Occ = FALSE Pointx.Position = Normal Pointx.Locked = FALSE Point1.Position = Normal Point1.Locked = FALSE Point2.Position = Normal Point2.Locked = FALSE Signalx.Colour = Red Signal1.Colour = Red	Signal2.Colour = Red Signal21.Colour = Red Train1.SituationID = ESx Train1.NextSituationID = ESx Train1.Request = ES11 Train2.SituationID = ES2 Train2.NextSituationID = ES2 Train2.Request = ReqEnd Route1.Locked = FALSE Route1.Free = TRUE

Çizelge 5.3 Raydan çıkma durumu için karşı örnek(devamı)

-> State: 1.3 <-	Signal11.Colour = Red	
Pointx.Position = Inverse	Route9.Free = TRUE	
Signalx.Colour = Yellow	Route10.Locked = FALSE	
Train1.NextSituationID = TCx	Route10.Free = TRUE	
Route2.Locked = FALSE	Route11.Locked = FALSE	
Route2.Free = TRUE	Route11.Free = FALSE	
Route3.Locked = FALSE	InitTrack_1 = ESx	
Route3.Free = FALSE	InitTrack_2 = ES2	
Route4.Locked = FALSE	Request_1 = ES11	
Route4.Free = FALSE	Request_2 = ReqEnd	
Route5.Locked = FALSE		
Route5.Free = FALSE	**-> State: 1.4 <-**	
Route6.Locked = FALSE	Trackx.Occ = TRUE	
Route6.Free = FALSE	Antennax.Occ = FALSE	
Route7.Locked = FALSE	Pointx.Position = Normal	
Route7.Free = FALSE	Train1.SituationID = TCx	
Route8.Locked = FALSE	Train1.NextSituationID= TC11	
Route8.Free = TRUE		
Route9.Locked = FALSE		

Elde edilen karşı örnek analizi yapıldığında görülmektedir ki tren TCx konumundayken Px normal pozisyona gelmektedir. Bu durum trenin raydan çıkmasıyla sonuçlanacaktır. Raydan çıkma durumu araştırıldığında Rota1 in kilitlenmesinden sonra tren TCx e gelmiş fakat bir sonraki durumda Px e geçememiştir. Çünkü Px makasının pozisyonu yanlış ayarlanmıştır. Bu durumda sorunun Rota1 de olduğu açıkça anlaşılmaktadır. Bu örnekte hatanın kaynağını bilmeseydik bile görüldüğü üzere hatanın nerede olduğunu bulmak karşı örnek analizi ile mümkündür.

BÖLÜM 6

SONUÇ

Projenin ilk kısmında derin bir literatür taraması yapılmış ve anklaşman sisteminin model kontrolü için kullanılacak yöntemler belirlenmiştir. Modelleme ASM notasyonu ile gerçekleştirilmiş ve sistemin gereksinimlerini ifade etmek için CTL zamansal lojik ifadeleri kullanılmıştır.

Model kontrolü yazılımı olarak, akademik açıdan geniş bir kullanım alanına sahip olması ve sistemin ASM modelinden model kontrolü için hazırlanması gereken koda dönüştürülürken bilgi kaybına neden olmaması bakımından NuSMV model kontrolü yazılımının kullanılması uygun bulunmuştur.

Model kontrolü "50. Yıl Bastabya" topolojisi için yapılmış olup, genel olarak kullanılan elemanların bulunduğu bir anklaşman tablosu oluşturulmuştur. Bu şekilde ileride farklı topolojilerle yapılabilecek çalışmalar için bir model tabanı tasarlanmıştır. Anklaşman sisteminin model kontrolü sonucu hataların bulunması için elde edilen karşı örneğin analizinin nasıl kullanılabileceğinden bahsedilmiştir.

Çalışmanın ilk safhasında konvansiyonel raylı sistemler için tasarlanan anklaşman tablosunun model kontrolü yapılarak hızlı trenler için oluşturulacak modelin temeli atılmıştır. Konvansiyonel raylı sistemlerin anklaşman tablolarıın model kontrolünde karşılaşılan sorunlara çözüm getirilebildiğinden, sonraki aşamada hızlı trenler için tasarlanan anklaşman sistemlerinin model kontorlüne geçmek daha kolay olacaktır. Bu çalışma sonrasında modelde hız verisinin de tanımlanması ve bunun tren hareketine ve kontrol sistemine nasıl etkiyeceğinin belirlenmesi sonucu hızlı trenlerde de model kontrolünün sağlanması planlanmaktadır.

KAYNAKLAR

[1] http://www.tcdd.gov.tr/, 17 Haziran 2015.

[2] Goldberg, B.A., Mootha R.K. ve Lindsey R.,W. , (1998). "Train accidents involving pedestrians, motor vehicles, and motorcycles" Am J Orthop, 27(4): 315-20.

[3] Evans, A.W., (2000). "Fatal train accidents on Britain's mainline railways", J.R. Statist. Soc. A, 163, 99-119.

[4] Kuepper, G.J., (1999). "150 Years of Train-Disasters - Practical Approaches for Emergency Responders", 9-1-1 Magazine, September/October issue, 30-33.

[5] Yıldırım, U., Durmuş, M.S., Söylemez, M.T. , (2011). "Demiryolu Sinyalizasyon Sistemleri İçin Otomatik Anklaşman Tablosu Oluşturulması" , EUSİS 2011, Bursa-Eskişehir, Türkiye.

[6] Clarke, E.M., Emerson E.A., (1981). "Design and synthesis of of synchronization skeletons using branching time temporal logic", Logic of Programs: Workshop, Sept 1981, Yorktown Heights, NY.

[7] Quielle, J.P., Sifakis, J., (1981). "Specification and verification of concurrent systems in CESAR", in: Proceedings of the 5th international symposium on programming, 337–350.

[8] Clarke, E.M., (2008). "Birth of Model Checking", Department of Computer Science, 25 Years of Model Checking, 1-26, Springer-Verlag Berlin, Heidelberg.

[9] Lichtenstein, O., Pnueli, A. , (1985). "Checking that finite state concurrent programs satisfy their linear specification", Proc. 12th ACM Symp. Principles of Programming Languages, POPL ’85, New Orleans, LA, USA, 97–107.

[10] McMilllan, K. L., (1992). "An approch to state explosion problem", CMU-CS, 92-131.

[11] Tian, C. , Duan Z., Zhang N., (2012). "An efficient approach for abstraction-refinement in model checking", Theoretical Computer Science, 46: 76-85.

[12] Hoare, C. A. R., (1985). Communicating Sequential Processes, Prentice-Hall International, London.

[13] Winter, K., (2002). "Model Checking Railway Interlocking Systems", M. Oudshoorn, Proc. of Australian Computer Science Conference (ACSC 2002), Australian Computer Science Communications, Australia.

[14] Simpson, A. , (1998). "Model Checking for Intrelocking Safety", In Proceeding of the Second FMERail Seminar.

[15] Winter, K., Neil J. Robinson, (2003). "Modelling Large Interlocking Systems and Model Checking Small Ones", M. Oudshoorn , 26th Australasian Computer Science Conference, Australian Computer Science Communications, Australia.

[16] Gurevich, Y., (1995). Evolving Algebras, Lipari Guide, E.Börger ed. Specification and Validation Methods, Oxford Universtiy Press.

[17] Cimatti, A., Giunchiglia, F., Mongardi, G., Romano, D., Torielli, F. ve Traverso, P., (1998). "Formal Verication of a Railway Interlocking System using Model Checking" , Formal Aspects of Computing (1998) 10: 361-380.

[18] Anunchai, S.V., (2009). "Verification of Railway Interlocking Tables using Coloured Pertri Nets", Proceedings of the 10th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools.

[19] Jensen, K. , (1987). "Coloured Petri nets" ,Petri Nets: Central Models and Their Properties Lecture Notes in Computer Science, 248-299.

[20] Emerson, E.A., (1995). Temporal and Modal Logic - Handbook of Theoretical Science,Elseiver, Amsterdam.

[21] Baier, C., Katoen, J.P., (2008). Principles of Model Checking, MIT Press.

[22] http://www-i2.informatik.rwth-aachen.de/Teaching/Course/MC/2005, 23 Ağustos 2015.

[23] http://cse.csusb.edu/dick/maths/logic_9_Modalities.html, 28 Ağustos 2015.

[24] Börger, E., Stark, R., (2003). "Abstract State Machines: A Method for High-Level System Design and Analysis", Springer, Verlag.

[25] Arcaini, P., (2012). Tool-Assisted Validation and Verification Techniques for State-Based Formal Methods , University of Milan, Informatics Department, Milan.

[26] Lewis, M.J.T., (2003). “Railways in the Greek and Roman World.”, Early Railways: A Selection of Papers from the First International Early Railways Conference, London.

[27] Theeg, G., Vlasenko, S., (2009). “Railway Signalling & Interlocking”, DVV Media Group GMBH-Eurailpress , 39-112.

[28] http://nusmv.fbk.eu/, 11 Eylül 2015.

[29] Öz, M. A. N., Şener, İ., Kaymakçı, Ö. T., Üstoğlu, İ., ve Cansever, G. (2015). "Topology Based Automatic Formal Model Generation for Point Automation Systems" Information Technology and Control, 44(1): 98-111.

[30] Şener, İ., Kaymakçı, Ö. T., Üstoğlu, İ., ve Cansever, G. (2014), "Specification and Formal Verification of Safety Properties in Point Automation System by Using Timed-Arc Petri Nets", 19th IFAC World Congress.,May 2014, Cape Town, South Africa.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	:Basri Tuğcan ÇELEBİ
Doğum Tarihi ve Yeri	:14.02.1989 - Seyhan
Yabancı Dili	:İngilizce
E-posta	:tugcan89@gmail.com

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Yüksek Lisans	Kontrol Müh.	Yıldız Teknik Üni.	2016
Lisans	Elektrik Müh.	Yıldız Teknik Üni.	2012
Lisans	Matematik Müh.	Yıldız Teknik Üni.	2012
Lise	Fen Bilimleri	ÇEAŞ Anadolu Lisesi	2007

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2015	ARTI Otomasyon	Proje Mühendisi
2014	NOKTA End. Otomasyon	Proje Mühendisi
2013	DAF Enerji	Ar-Ge Mühendisi