



**BOND GRAF TEKNİĞİNİ KULLANARAK DEVRELERİN
MODELLENMESİ VE SİMÜLASYONU İÇİN BİR
BİLGİSAYAR YAZILIMININ GELİŞTİRİLMESİ**

Fatma MURAT

Yüksek Lisans Tezi

Devreler ve Sistemler Anabilim Dalı

Danışman: Prof. Dr. Yakup DEMİR

HAZİRAN-2019

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**BOND GRAF TEKNİĞİNİ KULLANARAK DEVRELERİN
MODELLENMESİ VE SİMÜLASYONU İÇİN BİR
BİLGİSAYAR YAZILIMININ GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Fatma MURAT

(161113105)

Tezin Enstitüye Verildiği Tarih: 13.05.2019
Tezin Savunulduğu Tarih: 10.06.2019

Tez Danışmanı : Prof. Dr. Yakup DEMİR (F.Ü.)
Diğer Jüri Üyeleri : Prof. Dr. Arif GÜLTEN (F.Ü.)
Dr. Öğr. Üyesi Özal YILDIRIM (M.Ü.)



HAZİRAN-2019

ÖNSÖZ

Bond Graf tekniđi sistemlerin modellenmesi amacıyla geliştirilmiş metotlardan birisidir. Bu metodla karmaşık yapılı sistemlerde, sistemdeki enerji alış veriş modeli üzerinde ilk bakışta açıkça görülmekte ve bu durum sistem tasarımcısına büyük kolaylıklar sağlamaktadır. Bond Graf modelinden yararlanan çok sayıda yazılım bulunmaktadır. Bu çalışmada, Bond Graf modelini kullanan yazılımlara alternatif bir yazılımın oluşturulması amaçlanmıştır.

Yüksek lisans tez çalışmam boyunca bilgi birikimi ve tecrübeleri ile bana yol gösteren ve destek olan danışman hocam Prof. Dr. Yakup DEMİR'e, ilgisi ve özverişiyle çalışmalarımda yardımcı olan Munzur Üniversitesi Bilgisayar Mühendisliği Bölümünden Dr. Öğretim Üyesi Özal YILDIRIM'a, tez çalışmamın başında göstermiş olduđu ilgiden dolayı Dr. Öğretim Üyesi Ulaş Baran BALOĞLU'na, manevi destekleriyle hayatıma pozitiflik katan arkadaşlarıma ve son olarak çalışmalarım boyunca maddi manevi destekleriyle beni hiçbir zaman yalnız bırakmayan aileme sonsuz teşekkürler ederim.

Fatma MURAT
ELAZIĞ – 2019

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	I
İÇİNDEKİLER	II
ŞEKİLLER LİSTESİ	III
TABLolar LİSTESİ	V
KISALTMALAR LİSTESİ.....	VI
ÖZET	VII
ABSTRACT.....	VIII
1. GİRİŞ	1
2. BOND GRAF TEKNİĞİ	7
2.1. Giriş	7
2.2. Tanımlar.....	7
2.3. BG Bileşenleri.....	9
2.3.1. Bir Kapılı Elemanlar	9
2.3.1.1. Direnç.....	9
2.3.1.2. Kondansatör	9
2.3.1.3. İndüktans.....	10
2.3.1.4. Gerilim ve Akım Kaynakları	10
2.3.2. İki Kapılı Elemanlar.....	10
2.3.3. Çok Kapılı Elemanlar	11
2.3.4. Bağımlı Kaynaklar	12
2.4. Nedensellik (Kozalite)	13
2.5. BGM'nin Elde Edilişi	16
3. FORMÜLASYON VE ÇÖZÜM.....	20
3.1. Başlangıç Denklemleri ve Durum-Çıkış Denklemleri.....	20
4. BONAN PROGRAMI	25
4.1. Giriş	25
4.2. BONAN Programının Yapısı.....	26
4.2.1. Bond Elemanları Menüsü	27
4.2.2. Pasif Devre Elemanları Menüsü	29
4.2.3. Aktif Devre Elemanları Menüsü.....	30
4.3. BONAN Programı Akış Şeması	33
5. ÖRNEK DEVRELER.....	37
5.1. Örnek 1	37
5.2. Örnek 2	42
5.3. Örnek 3	44
5.4. Örnek 4	46
5.5. Örnek 5	48
6. SONUÇLAR VE GELECEK ÇALIŞMALAR	50
KAYNAKLAR	51
EK -1	55
EK -2	58
ÖZGEÇMİŞ	60

ŞEKİLLER LİSTESİ

	<u>Sayfa No</u>
Şekil 2. 1. (a) İki alt sistem, (b) iki enerji kapısının Paynter gösterimi	8
Şekil 2. 2. Direnç elemanının BG yönteminde gösterimi	9
Şekil 2. 3. Kondansatör elemanının BG yönteminde gösterimi	9
Şekil 2. 4. İndüktans elemanının BG yönteminde gösterimi	10
Şekil 2. 5. (a) Gerilim ve (b) Akım kaynaklarının BG yönteminde gösterimi	10
Şekil 2. 6. Transformatörün BG yönteminde gösterimi.....	11
Şekil 2. 7. Jiratorün BG yönteminde gösterimi	11
Şekil 2. 8. Seri çok kapılıının (a) BG gösterimi ve (b) Elektriksel karşılığı.....	12
Şekil 2. 9. Paralel çok kapılıının (a) BG gösterimi ve (b) Elektriksel karşılığı	12
Şekil 2. 10. Bağımlı kaynakların BG yönteminde gösterimi (a) Gerilim kontrollü gerilim kaynağı (b) Akım kontrollü gerilim kaynağı (c) Gerilim kontrollü akım kaynağı (d) Akım kontrollü akım kaynağı	13
Şekil 2. 11. Nedenselliğin tanımı ve gösterimi	13
Şekil 2. 12. (a) Çaba ve (b) Akış kaynaklarının nedensellik gösterimi	14
Şekil 2. 13. (a) Transformatör ve (b) Jirator elemanlarının nedensellik tanımları	14
Şekil 2. 14. (a) 0 ve (b) 1-kapılarının nedensellik tanımları	14
Şekil 2. 15 a) Örnek devre, b) BGM oluşturma 1. adım.....	16
Şekil 2. 16. BGM oluşturma 2. adım	17
Şekil 2. 17. BGM oluşturma 3. Adım	17
Şekil 2. 18. BGM oluşturma 4. Adım	18
Şekil 2. 19. BGM oluşturma 5. Adım	18
Şekil 2. 20. BGM oluşturma 6. Adım	19
Şekil 2. 21. BGM oluşturma 7. Adım	19
Şekil 4. 1. LogicSim hakkında genel bilgiler.....	26
Şekil 4. 2. BONAN program açılış sayfası.....	27
Şekil 4. 3. Bond elemanları menüsü	27
Şekil 4. 4. Bond elemanlarına ait özellikler kısmı.....	28
Şekil 4. 5. Bond kapı eşitliği hatasının gösterimi	28
Şekil 4. 6. Pasif devre elemanları menüsünün gösterimi	29
Şekil 4. 7. Direnç elemanının özellikler kısmı.....	29
Şekil 4. 8. İndüktans elemanının özellikler kısmı.....	30
Şekil 4. 9. Kondansatör elemanının özellikler kısmı	30
Şekil 4. 10. Kaynaklar menüsünün gösterimi	30
Şekil 4. 11. Gerilim kaynağının özellikler kısmı	31
Şekil 4. 12. Akım kaynağının özellikler kısmı	31
Şekil 4. 13. BONAN programının genel akış diyagramı	33
Şekil 4. 14. Düğüm matrisi oluşturma alt akış diyagramı	34
Şekil 4. 15. Düğüm matrisi içerisinde dal ve giriş elemanlarını seçme alt diyagramı....	35
Şekil 4. 16. Dal ve giriş katsayı matrislerinden, temel çevre ve temel kesitleme matrislerini elde etme alt diyagramı	36
Şekil 5. 1. Örnek 1'e ilişkin (a) devre şeması ve (b) BG modeli	37
Şekil 5. 2. Örnek 1'e ilişkin uygun bond seçimi	37
Şekil 5. 3. Elemanlara ilişkin özellikleri seçme menüsü	38
Şekil 5. 4. Örnek 1'e ilişkin bond özellikleri.....	38
Şekil 5. 5. Örnek 1'e ilişkin kaynak verileri.....	38

Şekil 5. 6. Örnek 1'e ilişkin direnç verileri	39
Şekil 5. 7. Örnek 1'e ilişkin indüktans verileri.....	39
Şekil 5. 8. Örnek 1'e ilişkin kondansatör verileri.....	39
Şekil 5. 9. Örnek 1'e ilişkin BONAN program çıktıları.....	40
Şekil 5. 10. Örnek 1'e ilişkin indüktans akımının değişimi	41
Şekil 5. 11. Örnek 1'e ilişkin kondansatör gerilimi değişimi.....	41
Şekil 5. 12. Örnek 2'ye ilişkin kaynak verileri.....	42
Şekil 5. 13. Örnek 2'e ilişkin BONAN program çıktıları.....	42
Şekil 5. 14. Örnek 2'e ilişkin indüktans akımının değişimi	43
Şekil 5. 15. Örnek 2'e ilişkin kondansatör gerilimi değişimi.....	43
Şekil 5. 16. Örnek 3'e ilişkin devre şeması	44
Şekil 5. 17. Örnek 3'e ilişkin BONAN program çıktıları.....	44
Şekil 5. 18. Örnek 3'e ilişkin indüktans akımı değişimi	45
Şekil 5. 19. Örnek 3'e ilişkin kondansatör gerilimi değişimi.....	45
Şekil 5. 20. Örnek 4'e ilişkin devre şeması	46
Şekil 5. 21. Örnek 4'e ilişkin BONAN program çıktıları.....	46
Şekil 5. 22. Örnek 4'e ilişkin indüktans akımı değişimi	47
Şekil 5. 23. Örnek 4'e ilişkin kondansatör gerilimi değişimi.....	47
Şekil 5. 24. Örnek 5'e ilişkin devre şeması	48
Şekil 5. 25. Örnek 5'e ilişkin BONAN program çıktıları.....	48
Şekil 5. 26. Örnek 5'e ilişkin kondansatör geriliminin değişimi.....	49

TABLÖLAR LİSTESİ

Sayfa No

Tablo 2. 1. Bond Graf değişkenlerinin çeşitli enerji domenilerindeki karşılığı.....	8
Tablo 2. 2. BG bileşenlerinin gösterimi, nedensellik ve matematiksel tanımları	15



KISALTMALAR LİSTESİ

BG	: Bond Graf
BGM	: Bond Graf Modeli
FDI	: Fault Diagnostic and Isolation (Hata Tespiti ve İzolasyonu)
ARR	: Analytical Redundancy Relations (Analitik Artık İlişkiler)
WBGos	: Word Bond Graf Objects (Kelime Bond Graf Nesneleri)
IDE	: Integrated Development Environment (Entegre Geliştirme Ortamı)

ÖZET

BOND GRAF TEKNİĞİNİ KULLANARAK DEVRELERİN MODELLENMESİ VE SİMÜLASYONU İÇİN BİR BİLGİSAYAR YAZILIMININ GELİŞTİRİLMESİ

Bu tez çalışmasında, elektrik devrelerinin modellenmesi ve analizi için Bond Graf (BG) yöntemini kullanan BONAN (Bond Graf Analiz) isimli bir bilgisayar programının oluşturulması amaçlanmıştır. Görsel kullanıcı arayüzüne sahip bu bilgisayar programında, devrelere ait Bond Graf Modeli (BGM) kullanıcı tarafından oluşturulmaktadır. Ayrıca, arayüz sayesinde devre elemanlarına ait bilgiler kolayca girilebilmektedir.

Bu tezde oluşturulan bilgisayar programı, Netbeans geliştirme ortamında Java programlama dili ile yazılmıştır. Programdan elde edilen veriler, Matlab ortamında yazılan çözüm algoritmasıyla işlenmiş ve grafiksel sonuçlar elde edilmiştir. Matlab ortamının özellikle çözüm kısmındaki üstünlükleri, oluşturulan programda yazılım sürecinin kısılmasını sağlamıştır. Programdan elde edilen sonuçlardan, programın doğruluğu görülmüştür.

Sonuç olarak, program BG modelini kullanarak gerçekleştirilen analiz aşamalarında başarılı bir yöntemdir. Ayrıca, farklı yazılımlar için örnek teşkil edecek özelliğe geliştirilebilme potansiyeline de sahiptir.

Anahtar Kelimeler: Bond Graf Modeli, Arayüz Programı, Formülasyon ve Çözüm.

ABSTRACT

DEVELOPING A COMPUTER SOFTWARE FOR MODELING AND SIMULATION OF THE CIRCUITS USING BOND GRAPH TECHNIQS

In this thesis, a computer program called BONAN (Bond Graph Analysis) which uses Bond Graph (BG) method for the modeling and analysis of electrical circuits is developed. In this computer program, Bond Graph Model (BGM) of the circuits is created by the users via an easy to use visual interface. In addition, information about circuit elements can be easily entered via this interface.

Java programming language and Netbeans development environment are used to create BONAN software. The data obtained from the program is processed with the solution algorithm written in Matlab environment and graphical results are obtained. The main advantage of the using Matlab environment, especially in the solution part, enabled the software tasks to be reduced. It is observed that BONAN software analyzes electrical circuits accurately.

As a result, developed computer program is a successful method in the analysis stages using the BG models. It also has the potential to develop an exemplary feature for different software.

Keywords: Bond Graph Method, Interface Program, Formulation and Solution.

1. GİRİŞ

Mühendislik alanında dinamik sistemlerin modellenmesi ve formülasyonu, sistemin analizi için gerekli bir adımdır. Sistemleri modellemek için çeşitli yaklaşımlar bulunmaktadır [1,2,3]. Grafiksel yaklaşım metotlarından bazıları; blok diyagramları, işaret akış diyagramları ve Bond Graf (BG) tekniğidir. Blok diyagramları ve işaret akış diyagramları denklemlerin bir tasvirini sağlar, operatörlerin hareketlerini işaretler üzerinden temsil eder. Fakat, sistemin topolojisi hakkında çok fazla fikir vermezler [4]. BG tekniği sistemlerin modellenmesi amacıyla geliştirilen metotlardan birisidir. Klasik graf teknikleri ile BG tekniği arasında önemli benzerlikler olmakla beraber, BG tekniğinin üstünlüğü, herhangi bir karmaşık yapıyı fiziksel sistem modelinin geometrik ve gözleme dayanan bir yolla doğrudan sisteme bakılarak elde edilebilmesidir [5]. İşaret-akış diyagramlarıyla modellenen bir sistemin iç ve uç değişkenleri arasındaki lineer fonksiyonel graf yapısı yerine, BG’da fiziksel bileşenler ve bir fiziksel bileşenin bağlı olduğu enerji kapıları arasında güç alış-verişi üzerine fonksiyonel bir geometri kurulmuş olur. Ayrıca, enerji kapılarının başka bir kapı ile kontrol edilmesi durumunda da BG tekniği avantajlıdır [3].

BG tekniğiyle modellemenin temeli, bir sisteme güç ve enerji akışıdır. Bu yüzden çoklu ortamlarda kesintisiz entegrasyon sağlar ve farklı ortamlar birleşik bir ortamda temsil edilebilirler [4]. Elektromekanik, elektro termal veya termodinamik sistemler birleşik notasyonlar ve tekniklerle birlikte tasvir ve analiz edilebilir. Örneğin, elektro-mekanik bir sistem için elektriksel ve mekanik alt sistemlerin ayrı ayrı analizini yapmaksızın, bütün bir sistem olarak analiz edilebilmesini sağlar. Bunun yanı sıra biyolojik ve kimyasal süreçler de, BG aracılığıyla işlenmiştir [6]. Ayrıca sistemlerdeki enerji alış verişi model üzerinde ilk bakışta açıkça görülür, bu durum sistem tasarımcısına kolaylık sağlar.

Devre teorisi ve graf yapılarının temeli 19. yüzyıl başlarında George Ohm [7,8,9] ve Gustav Kirchhoff [10,11] tarafından ortaya çıkmış günümüze kadar gelişmiş, özelleştirilmiştir. 1870’lerde kimyasal yapı grafları İngiliz matematikçi William Kingdom Clifford’un dikkatini çekmiştir. Clifford 1879 yılındaki vefatına kadar topolojiksel-matematiksel kavramlar olan “lineer graf”, “düğüm” ve “dal”

kavramlarıyla ilgilenmiştir [12,13]. Peirce, verimli bir uygulama olarak “Pierce’nin ilişkiler mantığı” ilişki sisteminin topolojisiyle ilgilenmiştir. Peirce’nin ilişkilere dair graflar kullanması, 1886 yılında Kempe tarafından dikkat çekici bir makale yayınlanmasının ardından fazla ilgi görmüştür [14,15,16].

BG günümüzdeki haliyle ilk olarak 1959 yılında Prof. Henry M. Paynter tarafından oluşturulmuştur. Paynter, BG’ı ilk olarak bir hidroelektrik santralini modellemede kullanmıştır. Hidroelektrik santralindeki elemanlar arasındaki güçlü analogilerin varlığı, Paynter’i yeni bir temsil şekli bulmaya yönlendirmiştir. Benzer analogiler, bir takım ortak genellemeyi ortaya çıkarmıştır. Analog cihazlar arasındaki benzerlik, bilgisayar simülasyonu için denklemlere indirgendiğinde aralarındaki farklar kaybolmuştur. Matematikçi Eliakim Hastinh Moore “Çeşitli teorilerin merkezi özellikleri arasındaki analogilerin varlığı, belirli teorilerin temelinde yatan ve onları bu merkezi özelliklere göre birleştiren genel bir teoremin varlığına” işaret eder [17]. Aralarındaki güçlü benzerlilerin bulunduğu bu tür elemanların varlığı, 0-1 jonksiyon kapıları fikrinin doğmasına yol açmıştır. 1959 yılında Paynter, 0-1 sayılarını, devre karmaşıklığını mantıksal ikiliğe indirgemek için sembol olarak seçmiştir. 1960 yılında MIT’de, bu sistemin temelini öğrencilere ve dışardan gelen katılımcılara anlatmıştır. BG sistemi, 1960 yılında Moscow IFAC’da tanıtılmış ve 1961 yılında MIT basın kitabı olan “Analysis and Design of Engineering System”de yayınlanmıştır [2,18].

Paynter’dan sonra BG yöntemi bazı araştırmacılarla mekanik sistemlere, akışkan dinamiğine, tersinir olamayan termodinamik sistemler ve benzeşimlerine uygulanmıştır. 1974 yılında Dean C. Karnopp ve Ronald C. Rosenberg tarafından sistem topolojisini temsil etme anlamına gelen BG’a bağlı ilk yazılı metin olan “System Dynamics; A Unified Approach” kitabı yayınlanmıştır [6]. Dean C. Karnopp [19,20,21], Ronald C. Rosenberg [22,23] dışında Alan J. Blundell [24], Peter C. Breedveld [25], Geneviève Dauphin-Tanguy [26], François Edouard Cellier [27,28], José J. Granda [29], Jean U. Thoma [30,31] gibi isimler de bu konuda çalışmalar yapmıştır. Ülkemizde ilk çalışmayı Prof. Dr. Necdet Şen [32] yapmış ve BG’ı Türkçeye “bağlaç diyagramı” olarak çevirmiştir. Daha sonra, Prof. Dr. Mustafa Poyraz [3] medikal sistemlere ilk uygulayan kimse olmuştur.

BG metodu çeşitli alanlarda kullanılmaktadır. 2014 yılında Roberto Tapia ve A. Medina, BG’ı çift beslemeli rüzgâr türbin tasarımında kullanarak sistemin sağlamlığını

doğrulanmış, sabit ve değişken rüzgâr hızı çalışma koşullarında simülasyonunu gerçekleştirmiştir. İndüksiyon makinasının Bond Graf Modeli (BGM) oluşturulmuş ve bu modele uygulanan nedensel döngü kavramı, model yaklaşımının dinamiğini tanımlamaya izin vermiştir [33].

Son yıllarda, BG'nin kullanıldığı sistemlerden biri de mekatronik sistemlerdir. Laurent Cauffriez ve arkadaşları 2015 yılında yayınladıkları makalede, yeni nesil bir tren kapısı mekatronik sisteminin arıza tespitini BGM kullanarak yapmışlardır. Bu sistem için referans modelini oluşturmada BG tekniği kullanılmıştır. Bu referans değere ek olarak Hata Tespiti ve İzolasyonuna (FDI, Fault Diagnostic and Isolation) dayalı genel bir model, kapı arızası durumunda hata göstergelerini ve artık eşiklerini üretmek için geliştirilmiştir. Önerilen bu mekatronik sistem, daha sonra test yatağında deneysel ölçümler alınarak doğrulanmıştır [34].

Arıza tespiti, mekatronik sistemlerin yanı sıra çeşitli sistemler için de önemlidir. Örneğin; Analitik Artık İlişkiler (Analytical Redundancy Relations) kullanan kararsız bir hidrolik sistemde gerçek zamanlı olarak sayısal FDI tasarımı için Ibrahim Abdallah ve arkadaşları 2017 yılında bir çalışma yürütmüştür. Bu çalışma, doğrudan gerçek zamanlı olarak dinamik eşikleri olan artıklar üretmiştir. Bu teşhis cihazı, hem modellemeye hem de teşhiste analitik modelin ortadan kaldırılmasına izin veren saf grafiksel modelleme ve teşhis yaklaşımı olan BGM'ni kullanır. Buradaki yenilikçi görüş, davranış modellenmesi ve FDI için, modelin analitik şeklini veya eşiklerini ifade etmeye gerek duymayan tek bir araç kullanılmasıdır [35].

BGM'nin farklı alanlarda kullanılmasına örnek olarak J.A. Tenreiro Machado ve Maria Eugenia Mata'nın 2014 yılında yayınlanan çalışmasında dünya ekonomisinin BG perspektifinden incelendiği görülmektedir. Dinamik sistemlerin teorisinden ve Brewer'ın 1975 yılında BG konseptini mikroekonomiye uyarlayan çalışmalarından ilham alınarak bu çalışmada bir BGM oluşturulmuştur. Günümüz küreselleşme sürecindeki ülkelerin potansiyel performanslarının iyi bir mikroekonomik temsili için düğüm ve bond olarak tanımlanan iki ana değişkeni baz alan bir dizi bağlantı ve dinamiksel bakış açısıyla oluşturulan bir model amaçlanmıştır. Geliştirilen yaklaşım, karmaşık dünyamızın bir sistem gibi nasıl hareket ettiğini algılamada yardımcı olmaktadır. Ayrıca, gelecekte dünyanın nasıl olacağı hakkında belirli bir bakış açısı oluşturmaya yardımcı olur. Son 50 yılda 50 ülke için zaman serilerine dayanan

tahminler, modelin doğruluğunu ve ölçeğin ekonomik performans için önemini kanıtlamıştır [36].

Biyomedikal sistemler için de BGM tasarımları mevcuttur. Anand Vaz, Kanwalpreet Singh ve Genevieve Dauphin-Tanguy 2014 yılındaki çalışmalarında insan parmağının ekstansiyon (extensor) mekanizmasının BGM'ni geliştirerek simülasyonunu sağlamışlardır. Parmak yapısındaki dinamik falanjlar (phalanges), kanca-dize etkileşimi (hook-string interaction), normal reaksiyon ve sürtünme kuvvetleri için Kelime Bond Graf Nesneleri (WBGOs, Word Bond Graf Objects) geliştirilmiştir. Tendonlardaki hareket ve kasılmaların incelenmesi, kancaların yeri ve bireysel tendonların karakteristikleri rahatlıkla BGM'ne göre yapılabilmektedir. Bu durum, bilgisayar simülasyonu ile etkin bir şekilde gösterilmiştir. Geliştirilen bu uygulama, cerrahi işlemlerden veya diğer nedenlerden dolayı tendon topolojisinde meydana gelen değişimlerin etkileri üzerine yapılan ileri çalışmalarda yardımcı olacaktır [37].

BG yapısının bilgisayar simülasyonu için durum denklemlerinin çıkarımını algoritmik yolla sağlaması, temel bir modelleme aracı olarak kullanılarak yazılımların gelişmesini sağlamıştır. Sembolik formda durum denklemlerini elde ettikten sonra tasarımcı, sistemin analizi ve tasarımı için bunları kullanabilir. BGM'ni kullanan simülasyona ilişkin çok sayıda yazılım vardır:

1. ENPORT: BG ve simülasyon yazılımının ilki 70'li yılların başlarında yazılan ENPORT'tur [38]. Bu yazılım, kozaliteyi belirtmeyi gerektirmemekte ve topolojik giriş tanımı daha sonra çözülebilir bir dal giriş matrisine dönüştürülmektedir. MATLAB programı ile birlikte, çözüm denklemlerini belirlemektedir. Bu yazılım, BG teorisinin geliştirilmesinde ve uygulamalarında aracı olmuştur. Ancak paket, yeni nesil yazılım uygulamalarına uymamaktadır [39].

2. CAMP-G: BG'ı girdi olarak tanımlayan fiziksel modelleri kullanan mühendislere ve bilim adamlarına, mekatronik ve dinamik sistemlerin tasarlanmasında yardımcı olan bir yazılım paketidir. Mekaniksel, elektriksel, hidrolik, termik ve kontrol sistemleri bilgisayar grafiklerini birlikte kullanarak modellenebilmektedir. CAMP-G fiziksel ve kontrol sistemlerinin bilgisayar simülasyonlarını yapmak için MATLAB/Simulink, ACSL gibi dillerle arabirimler oluşturan bir model üretim aracıdır. Ancak, son işlemleri yapmak için büyük ölçüde dış yazılımlara bağlıdır [40,41].

3. SYMBOLS-2000: Nesne odaklı, aşamalı hibrit modelleme, simülasyon ve kontrol analiz yazılımıdır. BG, blok diyagramı ve denklem modelleri kullanarak model oluşturmada kullanıcılara imkan sağlar. Farklı mühendislik ve modelleme alanları için “kapsül” olarak adlandırılan çok sayıda gelişmiş alt modelleri mevcuttur. Bu kapsül, ana yapıya dahil edildiğinde uygun düzenlemeler ile birleştirilmektedir. Amalendu Mukherjee ve A.K. Samantaray’ın 2001 yılında yaptıkları çalışmayla Encapsulation olarak adlandırdıkları bir aracı SYMBOLS-2000 bünyesine dahil etmiştir. Encapsulation’ın en önemli katkısı; birçok özelleştirme, kalıtım ve tam soyutlama kullanarak var olandan daha fazla kapsül elde etmektir. Bu yazılım zarif görünümlü çağdaş bir Graphical User Interface (GUI, Grafiksel Kullanıcı Arayüzü) kullanır. Sayısal ve sembolik çözüm yetenekleri oldukça gelişmiştir. Bu yazılım Microsoft Developer Studio™ ön kurulumu gerektirmektedir. Büyük sistemlerin endüstriyel modellenmesinde ve araştırılmasında, bu yazılımın kullanılması tavsiye edilir [42,43].

4. 20-sim: Windows arka planında çalışan bir modelleme ve simülasyon programıdır. Bu ürün DOS temelli TutSim yazılımından sonra gelişen bir zaman-testli modelleme aracıdır. Sembolik diyagramları, BG’ı, blok diyagramlarını, denklem modellerini veya bunların herhangi bir kombinasyonlarını destekleyen dinamik sistemlerin ileri düzey modellenmesini ve simülasyonunu sağlayan bir pakettir. Bununla birlikte elektriksel, mekaniksel ve hidrolik sistemlerin veya aralarındaki kombinasyonların dinamik sistemlerinin davranışını simule edebilir. Geliştirilen sürümü MATLAB ile uyumludur ve Simulink etkileşimine izin verir. Marisol Delgado ve Cesar Pichardo yaptıkları çalışmada doğal gazın sıvılaştırılma işleminde kullanılan iki fazlı bir flaş ayırıcının modellenmesini, BGM’ni kullanarak 20-sim ve MATLAB yazılımlarında simülasyonunu gerçekleştirmişlerdir. Karmaşık diyagramların oluşturduğu denklemlerin tanımlanma zorunluluğu olmadığı için kullanılan flaşın dinamik davranışının üretilmesinde 20-sim’le birlikte MATLAB’ın kullanılması BGM’nin simule edilmesini oldukça kolaylaştırmıştır [44,45,46].

5. Modelica: OMOLA’nın birçok fikrinden esinlenmiş, nesne tabanlı bir modelleme ve simülasyon aracıdır ve büyük, karmaşık, hibrit özellikli fiziksel sistemler için kullanılır. Çok alanlı modelleme için uygundur. Örnek olarak; robotik mekatronik sistemler, otomotiv ve havacılık, mekaniksel, elektriksel, hidrolik ve kontrol sistemlerini içeren uygulamalar, süreç odaklı uygulamalar, elektrik gücünün dağıtımı

uygulamaları sayılabilir. Jan F. Broenink yaptığı bir çalışmada 20-sim'den Modelicaya aktarılan bir filtre tasarlayıp bir örnek üzerinde doğruluğunu kanıtlamıştır [47].

Bu yazılımlar dışında François E. Cellier ve Robert T. Mcbride, Dymola yazılımı için bir BG kütüphanesi geliştirmişlerdir. Dymola modelleme yapısına BG'ın yerleştirilmesi BG modelleme metodolojisinin ifade gücünü ve esnekliğini arttırmıştır. Bu çalışma basit bir pozisyon kontrol sistemi örneği ile doğrulanmıştır [48].

Genel olarak BG teorisine göre yapılan yazılımlarda model, fiziksel davranışı düzeyinde çözülür. Bu yaklaşım, klasik modelin aksine, bağımsız bileşenler için denklemlerin ilk önce oluşturulduğu daha sonra simülasyon planının kendi bazında türetildiği metodun tersine işlemdir. Sistemlerin BGM'den adım-adım çıkarılma işlemi kullanılarak sistem denklemleri elde edilebilir.

Bu tez çalışmasında geliştirilen BONAN (**BON**d graf **AN**aliz) programı, masaüstü uygulaması olarak çalışmakta ve nesne tabanlı bir modelleme ve simülasyon aracıdır. Program, kozaliteyi otomatik olarak atadığından kullanıcıya büyük kolaylık sağlamaktadır. Ayrıca, program durum ve çıkış denklemlerine elde etmektedir. Bu denklemler, MATLAB programı kullanılarak çözdürülmektedir. Arayüz programı, kullanıcının herhangi bir programlama dili bilmesini gerektirmeden model oluşturabilmesine imkan vermektedir.

Tez çalışması, 6 ana bölümden oluşmaktadır. Bölüm 1'de, BG yöntemine ait genel bilgilere ve literatür taramasına yer verilmiştir. Bölüm 2'de, BG metoduna ait tanımlar, BG bileşenleri, nedensellik kavramı ve adım adım BGM'nin çıkarılması ayrıntılı olarak anlatılmıştır. Bölüm 3'de formülasyon ve çözüm kısmı anlatılmıştır. Bölüm 4'de, bu tezde oluşturulan BONAN programının tanıtımı yapılmıştır. Programda devre elemanlarının kullanılması ve veri girişinin nasıl yapılması gerektiği örnekler ile Bölüm 5'de anlatılmıştır. Bu bölümde verilen örneklerin grafiksel çıktıları Matlab ortamında elde edilmiştir. Bölüm 6'da oluşturulan bilgisayar programı hakkında bilgilendirme yapılmış ve gelecekte yapılabilecek çalışmaların neler olabileceğinden bahsedilmiştir.

2. BOND GRAF TEKNIĞİ

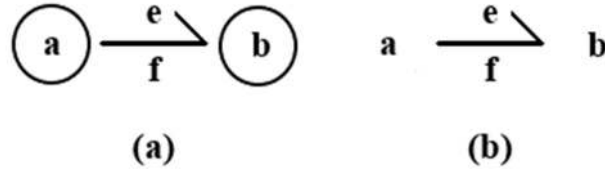
2.1. Giriş

BG Tekniğinin temeli, sistemin birbirine bağlı fiziksel bileşenleri arasındaki güç alış-verişi düşüncesidir. BG'la ilgili ilk çalışmayı, 1961'de M.I.T.'den Prof. H. M. Paynter yapmıştır. Paynter tarafından BG olarak adlandırılan bu düşünce, Prof. Dr. Necdet Şen tarafından Türkçe'ye "bağlaç diyagramları" ismiyle aktarılmıştır. BG tekniğinin sağladığı kolaylıklar aşağıdaki gibidir [2, 32]:

- Karmaşık yapılı bir sistem, kendisini oluşturan fiziksel bileşenlerin enerji elemanları ne olursa olsun aralarındaki enerji kapıları yardımıyla birleştirilmiş işleve uygun diyagramla modellenebilir.
- Verilen bir fiziksel sistemin bileşenleri arasındaki güç alış-verişini gösteren enerji kapılarının kurdukları diyagrama doğrudan bakarak, sistemin iç ve uç değişkenleri cinsinden bir takım matris-vektör bağıntıları kurulabilir. Bu bağıntılar, istenilen bir matematiksel modeli verecek şekilde BG'na bakarak elde edilebileceği gibi, bazı sistem fonksiyonlarının elde edilmesinde de kullanılabilir.
- Her bir BG bileşeni, ilgili olduğu fiziksel bileşenin iç ve uç değişkenini gösterdiğinden; sistem n tane BG bileşeni olan bir işaret akış diyagramına dönüştürülebilir.

2.2. Tanımlar

En temel BG bileşeni Şekil 2.1.'de gösterilen a ve b gibi iki alt sistem yada iki enerji kapısı arasına çizilen ve a'yı b'ye bağlayan bir çizgi parçasıdır. Bu gösterimin temelinde a'dan b'ye aktarılan bir $e \times f$ gücünün temsili yatmaktadır. Paynter tarafından tasarlanan bu gösterimde; e: çaba (uç) değişkeni, f: akış (iç) değişkenidir [2].



Şekil 2. 1. (a) İki alt sistem, (b) iki enerji kapısının Paynter gösterimi

Bir sistemde, elemanlar arasında mevcut olan güç akışı her iki yönde de olabilir. BG tekniğinde gücü oluşturan değişkenler, çaba ve akış değişkenleridir. Güç, bir kapının içine doğru veya kapıdan dışarı doğru olup, çaba ve akış değişkenlerinin çarpımı olarak ifade edilebilir. Şekil 2.1.'de, gücün pozitif yönü (akış yönü) gösterilmektedir. Burada, P güç olmak üzere a 'dan b 'ye $P=e \times f$ gücünün aktarıldığı gösterilmektedir. Dinamik bir sistemde, çaba ve akış değişkenleri dolayısıyla güç zamanla değişecektir. Bondun yani güç akışını gösteren okun yönü, e ve f 'nin yönlerinden bağımsızdır.

Tablo 2.1.'de çeşitli enerji domenindeki sistemlerin çaba ve akış değişkenleri gösterilmiştir. Elektrik devrelerinde çaba değişkeni gerilim, akış değişkeni ise akımdır. Bu tezde, elektriksel sistemlerle ilgilenildiğinden, bundan sonra BG'nin temel tanımlamaları yerine elektriksel karşılıkları kullanılacaktır.

Tablo 2. 1. Bond Graf değişkenlerinin çeşitli enerji domenilerindeki karşılığı

Enerji Domeni	Çaba	e sembolü	Metrik ölçüm sistemi	Imperial ölçüm sistemi	Akış	f sembolü	Metrik ölçüm sistemi	Imperial ölçüm sistemi
Elektrik	Elektromotor kuvveti	V, U	V	V	Akım	I	A	A
Mekanik Öteleme	Kuvvet	F	N	lb	Dogrusal Hız	v	m/s	ft/s,mph
Mekanik Dönme	Tork	τ	N.m	ft.lb	Açısal Hız	ω	rad/s	rad/s
Hidrolik	Basınç	P	Pa	psi	Debi	Q	m ³ /s	ft ³ /s
Termik	Sıcaklık	T	°C, °K	°F	Isı Akış Hızı	S	W/°C	ft.lb/s. °F

2.3. BG Bileşenleri

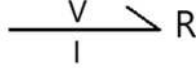
2.3.1. Bir Kapılı Elemanlar

BG yönteminde bir kapılı elemanlar; gerilim ve akım kaynakları ile direnç, indüktans ve kondansatördür. Bu elemanlar; enerji harcayan, enerji depolayan ve enerji veren olmak üzere sınıflandırılır.

2.3.1.1. Direnç

Direnç, akım ve gerilim değişkenleri arasında belirli bir ilişki kuran ve sistemdeki enerjiyi harcayan devre elemanıdır. BG yönteminde, tanımı (2.1)'de verilen direnç elemanının gösterimi Şekil 2.2.'de verilmiştir.

$$v(t) = R \cdot i(t) \quad (2.1)$$



Şekil 2. 2. Direnç elemanının BG yönteminde gösterimi

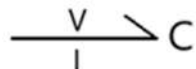
2.3.1.2. Kondansatör

BG yönteminde enerji depolayan kondansatör elemanının gösterimi Şekil 2.3'de verilmiştir. Kondansatör elemanının gerilim ve yük değişkenleri arasındaki bağıntısı (2.2)'de, akım ve gerilim değişkenleri arasındaki ilişkisi ise (2.3) ve (2.4)'de verilmiştir.

$$v = f_c(q) \quad (2.2)$$

$$v(t) = v(t_0) + \frac{1}{C} \cdot \int_{t_0}^t i(t) dt \quad (2.3)$$

$$\frac{dv(t)}{dt} = \frac{1}{C} \cdot i(t) \quad (2.4)$$



Şekil 2. 3. Kondansatör elemanının BG yönteminde gösterimi

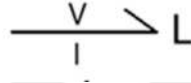
2.3.1.3. İndüktans

Enerji depolayan bir diğer eleman olan indüktansın BG gösterimi Şekil 2.4’de verilmiştir. İndüktans elemanının akım ile akı değişkenleri arasındaki ilişki (2.5)’de, akımı ile gerilimi arasındaki ilişkisi ise (2.6) ve (2.7)’de verilmiştir.

$$i = f_L(\lambda) \quad (2.5)$$

$$i(t) = i(t_0) + \frac{1}{L} \cdot \int_{t_0}^t v(t) dt \quad (2.6)$$

$$\frac{di(t)}{dt} = \frac{1}{L} \cdot v(t) \quad (2.7)$$



Şekil 2. 4. İndüktans elemanının BG yönteminde gösterimi

2.3.1.4. Gerilim ve Akım Kaynakları

Bir kapılı elemanlardan olan akım ve gerilim kaynakları, sisteme enerji veren devre elemanlarıdır. İdeal bir akım kaynağı gerilim bileşeninden bağımsız düşünülerek I_k ; ideal bir gerilim kaynağı ise akım bileşeninden bağımsız düşünülerek V_k olarak gösterilmiştir. Gerilim ve akım kaynaklarının BG gösterimi Şekil 2.5’de verilmiştir.

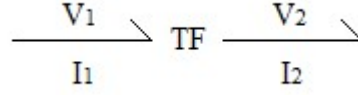


Şekil 2. 5. (a) Gerilim ve (b) Akım kaynaklarının BG yönteminde gösterimi

2.3.2. İki Kapılı Elemanlar

Birden fazla enerji modu bulunan karmaşık yapıli sistemlerde, aynı anda birden fazla enerji dönüşümü söz konusu olabilir. Enerji modları arasındaki enerji aktarımı sırasında sistem değişkenleri belli oranda aynı değişkenlere dönüşüyorsa, böyle bir elemana fiziksel transformatör denir. Transformatör elemanının BG yönteminde gösterimi Şekil 2.6’da verilmiştir.

Transformatör elemanına ilişkin tanım bağıntısı (2.8)’de verilmiştir. Burada, m : dönüştürme oranıdır.

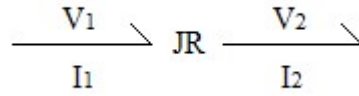


Şekil 2. 6. Transformatörün BG yönteminde gösterimi

$$\begin{bmatrix} V_1(t) \\ I_2(t) \end{bmatrix} = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \cdot \begin{bmatrix} V_2(t) \\ I_1(t) \end{bmatrix} \quad (2.8)$$

BG yönteminde, iki kapılı dönüştürücü olan bir diğer eleman fiziksel jiratördür. Sistem değişkenlerinin enerji aktarımında akım değişkenleri ile gerilim değişkenleri arasında çarpazlama bir değişim söz konusuysa bu tür bir değişim BG tekniğinde fiziksel jiratör elemanı ile ifade edilir. Jiratör elemanının BG yönteminde gösterimi, Şekil 2.7’ de verilmiştir.

Jiratöre ait tanım bağıntıları (2.9)’da verilmiştir. Burada, r : aktarım oranıdır.



Şekil 2. 7. Jiratörün BG yönteminde gösterimi

$$\begin{bmatrix} V_2(t) \\ I_2(t) \end{bmatrix} = \begin{bmatrix} 0 & r \\ 1/r & 0 \end{bmatrix} \cdot \begin{bmatrix} V_1(t) \\ I_1(t) \end{bmatrix} \quad (2.9)$$

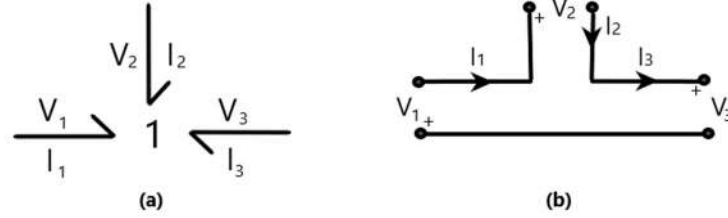
2.3.3. Çok Kapılı Elemanlar

BG yönteminde çok kapılı elemanlar, bağlantı elemanı olarak isimlendirilir. Seri (1-bağlantı elemanı, 1-kapısı) ve paralel (0-bağlantı elemanı, 0-kapısı) çok kapılılar olmak üzere 2 çeşittir.

Seri çok kapılılar BG tekniğinde “1” veya “s” ile ifade edilmiştir. Seri çok kapılıda akım değişkenleri birbirine eşit olup, kapıya bağlı elemanların gerilim değişkenleri toplamı sıfırdır. Seri çok kapılılara ilişkin bu eşitlik ve denklikler Şekil 2.8’de görülebilir. n-tane elemanın bağlı olduğu seri çok kapılılar için (2.10) ve (2.11)’deki bağıntılar geçerlidir.

$$i_1(t) = i_2(t) = \dots = i_n(t) \quad (2.10)$$

$$V_1(t) + V_2(t) + \dots + V_n(t) = 0 \quad (2.11)$$

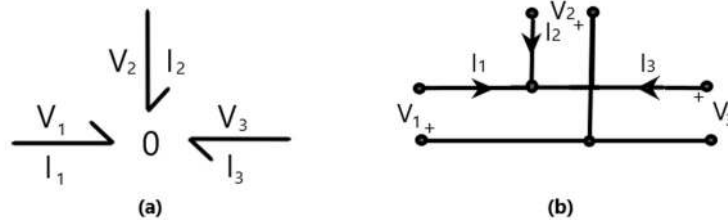


Şekil 2. 8. Seri çok kapılının (a) BG gösterimi ve (b) Elektriksel karşılığı

Paralel çok kapılı elemanda ise kapıya bağlı elemanların gerilim değişkenleri birbirine eşitken akım değişkenlerinin toplamı sıfırdır. BG yönteminde “0” veya “p” ile ifade edilirler. Paralel çok kapılılara ilişkin bu eşitlik ve denklikler, Şekil 2.9’de görülebilir. n-tane elemanın bağlı olduğu seri çok kapılılar için (2.12) ve (2.13)’deki bağıntılar geçerlidir.

$$V_1(t) = V_2(t) = \dots = V_n(t) \quad (2.12)$$

$$I_1(t) + I_2(t) + \dots + I_n(t) = 0 \quad (2.13)$$

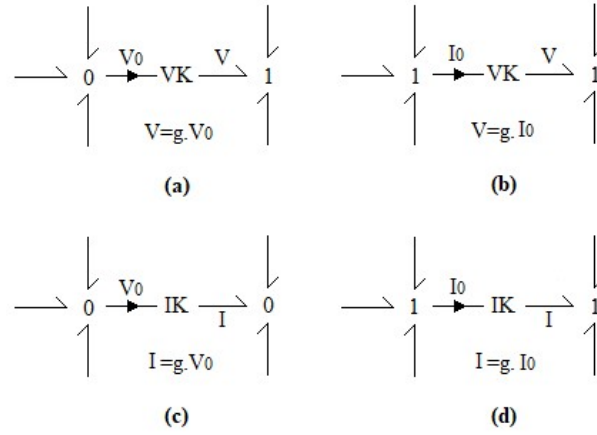


Şekil 2. 9. Paralel çok kapılının (a) BG gösterimi ve (b) Elektriksel karşılığı

Bu tezde seri çok kapılılar 1-kapısı, paralel çok kapılılar ise 0-kapısı olarak ifade edilecektir.

2.3.4. Bağımlı Kaynaklar

Bağımlı kaynaklar, devredeki herhangi bir gerilim veya akımın etkilediği türden kaynaklardır. Şekil 2.10’da bağımlı gerilim ve bağımlı akım kaynaklarının BG yönteminde gösterimi verilmiştir.

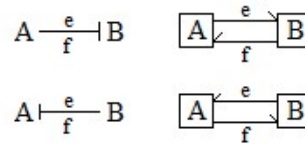


Şekil 2. 10. Bağımlı kaynakların BG yönteminde gösterimi (a) Gerilim kontrollü gerilim kaynağı (b) Akım kontrollü gerilim kaynağı (c) Gerilim kontrollü akım kaynağı (d) Akım kontrollü akım kaynağı

2.4. Nedensellik (Kozalite)

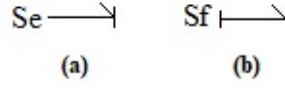
Sebep-sonuç ilişkisinin nedensellik özelliği kullanılarak bond üzerinde gösterilmesiyle oluşan diyagramda; herhangi bir yerdeki niceliğin, diyagramda başka herhangi yerdeki niceliği nasıl etkilediği belirtilir ve sebepten sonuca doğru bir nedensellik yolu izlenmiş olur.

Şekil 2.1'deki Paynter gösteriminde a'dan b'ye bir güç aktarımı söz konusudur. Fakat, çaba ve akış değişkenlerinin yönü de önemli bir faktördür ve diyagramda gösterilmesi gerekir. Nedensellik kavramı ile BG üzerinde bir bağlacın herhangi bir ucuna kısa ve dikey bir çizgi eklenerek bu eksiklik giderilmiştir. Çaba değişkeni nedenselliğe doğru, akış değişkeni ise nedensellikten dışarı doğru yönelir. Yani çaba ve akış değişkenlerinin yönleri daima birbirlerine zıt yöndedir. Bunun yanısıra nedensellikle, güç akışının yönü birbirinden bağımsızdır. Nedensellik tanımı Şekil 2.11'de gösterilmiştir.



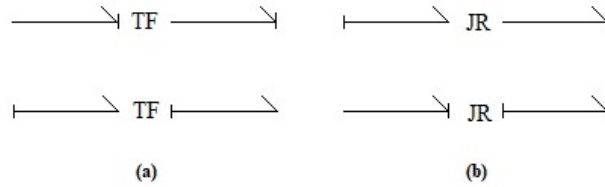
Şekil 2. 11. Nedenselliğin tanımı ve gösterimi

Çaba ve akış kaynaklarının nedensellik tanımları sabittir. Kaynak hangi tür değişkenle sistemi etkiliyorsa o değişkene göre nedenselliği belirlenir. Şekil 2.12’de Çaba ve akış kaynaklarının nedensellik tanımları ve gösterimi verilmiştir.



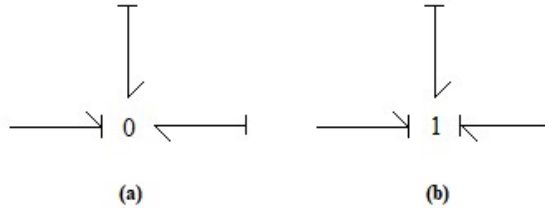
Şekil 2. 12. (a) Çaba ve (b) Akış kaynaklarının nedensellik gösterimi

Transformatör ve Jiratör gibi iki kapılı elemanların nedenselliği, iki çeşit olabilmektedir. Bu elemanlara ilişkin nedensellik tanımları ve gösterimi Şekil 2.13’de verilmiştir. Transformatör elemanı nedensellik yönünü aynen iletirken, jiratör elemanında nedensellik yönünü değiştirir. Yani jiratöre giren bir akış nedenselliği, çaba nedenselliği olarak çıkacaktır.



Şekil 2. 13. (a) Transformatör ve (b) Jiratör elemanlarının nedensellik tanımları

0 ve 1-kapılarının nedensellikleri, belirli kurallara göre tanımlanır. 0-kapısına bağlı herhangi bir çaba değişkeni giriş alınırsa diğer çaba değişkenleri çıkış alınmalıdır ve benzer şekilde 1-kapısına bağlı herhangi bir akış değişkeni giriş alınırsa diğer akış değişkenleri çıkış alınmalıdır. 0 ve 1-kapılarının nedensellik tanımları Şekil 2.14’de verilmiştir.



Şekil 2. 14. (a) 0 ve (b) 1-kapılarının nedensellik tanımları

Bir sistemin BGM oluşturulduktan sonra nedensellik tayini yapılırken, önce kaynakların nedensellikleri göz önünde bulundurulur. Daha sonra enerji depolayan elemanların nedenselliği yapılır. Direnç elemanının nedensellik seçimi serbesttir.

Bu tez çalışmasında program, elemanların nedenselliklerini formülasyon aşamasında otomatik olarak atamaktadır. Bütün BG elemanlarının gösterimi, nedensellikleri ve matematiksel ifadeleri Tablo 2.2’de verilmiştir.

Tablo 2. 2. BG bileşenlerinin gösterimi, nedensellik ve matematiksel tanımları

Bileşen	BG Gösterimi	Nedensellik	Matematiksel İfadeler
Çaba Kaynağı	$Se \longrightarrow$	$Se \longrightarrow \vdash$	$e(t) = E(t)$
Akış Kaynağı	$Sf \longrightarrow$	$Sf \vdash \longrightarrow$	$f(t) = F(t)$
Direnç	$\longrightarrow R$	$\longrightarrow \vdash R$ $\vdash \longrightarrow R$	$e(t) = R \cdot f(t)$
Kondansatör	$\longrightarrow C$	$\longrightarrow \vdash C$ $\vdash \longrightarrow C$	$e(t) = e(0) + \frac{1}{C} \cdot \int_{t_0}^t f(\tau) d\tau$
İndüktans	$\longrightarrow L$	$\longrightarrow \vdash L$ $\vdash \longrightarrow L$	$f(t) = f(0) + \frac{1}{L} \cdot \int_{t_0}^t e(\tau) d\tau$
Transformatör	$\longrightarrow TF \longrightarrow$	$\longrightarrow \vdash TF \longrightarrow \vdash$ $\vdash \longrightarrow TF \vdash \longrightarrow$	$\begin{bmatrix} e_1(t) \\ f_2(t) \end{bmatrix} = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \cdot \begin{bmatrix} e_2(t) \\ f_1(t) \end{bmatrix}$ $\begin{bmatrix} f_1(t) \\ e_2(t) \end{bmatrix} = \begin{bmatrix} 1/m & 0 \\ 0 & 1/m \end{bmatrix} \cdot \begin{bmatrix} f_2(t) \\ e_1(t) \end{bmatrix}$
Jirator	$\longrightarrow JR \longrightarrow$	$\longrightarrow \vdash JR \vdash \longrightarrow$ $\vdash \longrightarrow JR \longrightarrow \vdash$	$\begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} = \begin{bmatrix} 0 & r \\ r & 0 \end{bmatrix} \cdot \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix}$ $\begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1/r \\ 1/r & 0 \end{bmatrix} \cdot \begin{bmatrix} e_2(t) \\ e_1(t) \end{bmatrix}$
0-kapısı	$\longrightarrow 0 \longleftarrow$	$\vdash 0 \longleftarrow \vdash$	$e_1(t) = e_2(t) = e_3(t)$ $f_1(t) + f_2(t) + f_3(t) = 0$
1-kapısı	$\longrightarrow 1 \longleftarrow$	$\longrightarrow 1 \longleftarrow$	$f_1(t) = f_2(t) = f_3(t)$ $e_1(t) + e_2(t) + e_3(t) = 0$

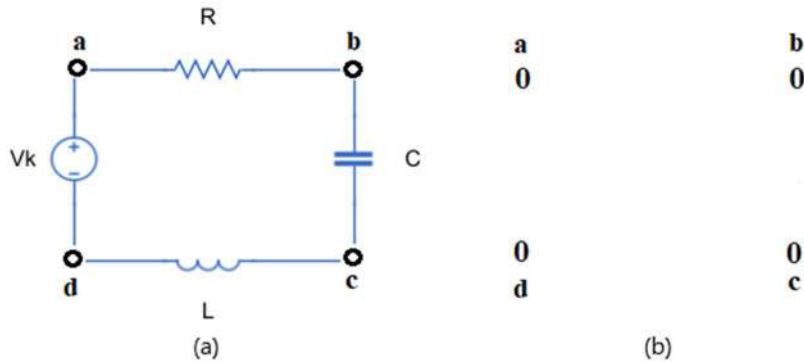
2.5. BGM'nin Elde Edilişi

Dinamik bir sistemin BGM oluşturulurken, önce sistemdeki bütün elemanlar Tablo 2.2'de verilen BG karşılıkları ile ifade edilirler. Sonra devre elemanları, akım veya gerilim değişkenlerinin durumuna göre 0 veya 1-kapılarıyla birbirine bağlanırlar. Daha sonra her bir kapı için nedensellikler belirlenir.

Bir devrenin BGM oluşturulurken, devredeki farklı potansiyellere sahip düğüm noktalarının yerine 0-kapıları yerleştirilir. 0-kapılarının aralarına 1-kapıları yerleştirilerek, devre elemanları (R, L, C, V_k, I_k) bondlar yardımıyla 1-kapılarına bağlanır. Bondların yönü tayin edildikten sonra, devrede öngörülebilir sıfır potansiyeline sahip 0-kapısı birlikte bağlı olduğu bondlar ile devreden çıkartılır.

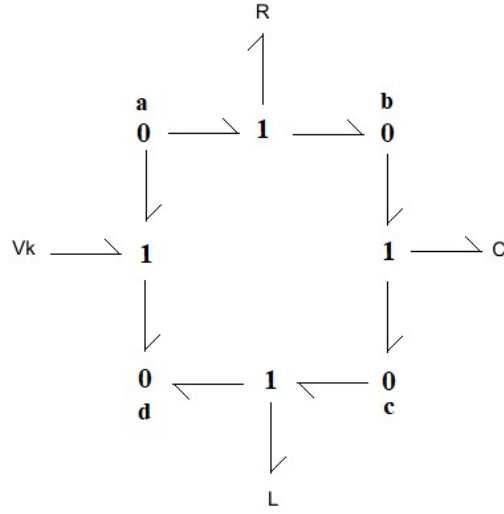
Bir BGM'nin oluşum adımları, Şekil 2.15.a'da verilen örnek bir seri RLC devresi üzerinde anlatılmıştır:

- **Adım 1:** Devredeki farklı potansiyellere sahip düğümler yerine Şekil 2.15.b'de belirtildiği şekliyle diyagramda 0-kapıları oluşturulur.



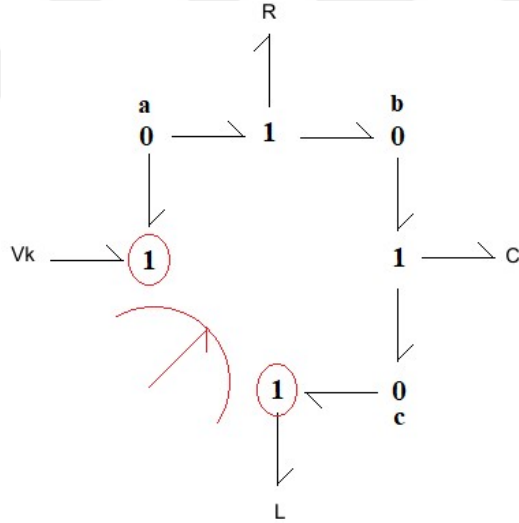
Şekil 2. 15 a) Örnek devre, b) BGM oluşturma 1. adım

- **Adım 2:** 0-kapılarının aralarına elemanlara bağlı 1-kapıları yerleştirilir (Şekil 2.16).



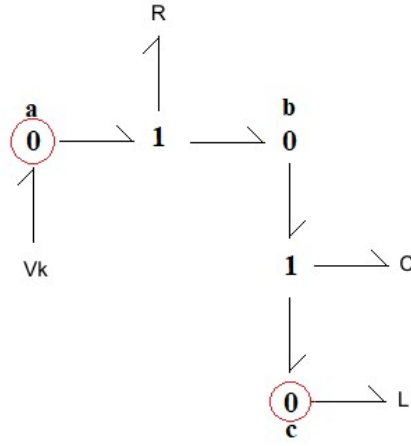
Şekil 2. 16. BGM oluşturma 2. adım

- **Adım 3:** Referans düğümü olarak, uygun bir 0-kapısı seçilerek diyagramdan çıkartılır (Şekil 2.17).



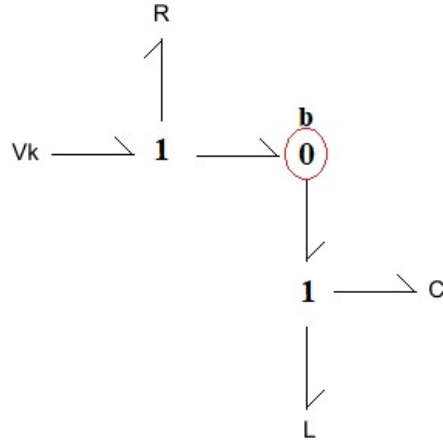
Şekil 2. 17. BGM oluşturma 3. Adım

- **Adım 4:** Diyagramın her iki ucunda kalan ve sadece bir eleman bağlı 1-kapıları kaldırılır. Elemanları ise ilgili 0-kapılarına bağlanır (Şekil 2.18).



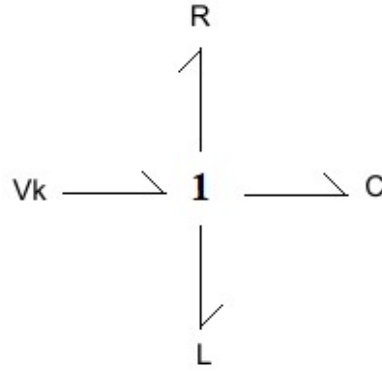
Şekil 2. 18. BGM oluşturma 4. Adım

- **Adım 5:** Diyagramın uçlarında kalan ve sadece bir elemanın bağlı olduğu 0-kapıları kaldırılarak, elemanları ilgili 1-kapılarına bağlanır (Şekil 2.19).



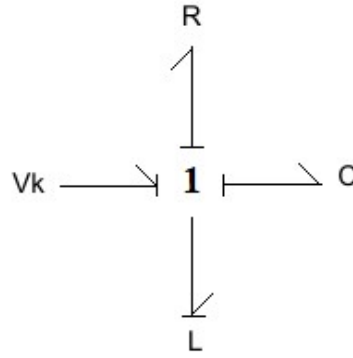
Şekil 2. 19. BGM oluşturma 5. Adım

- **Adım 6:** Diyagramın son halinde hiçbir elemanın bağlı olmadığı herhangi bir kapı varsa, bu da diyagramdan çıkarılır. İki tane 1-kapısı veya iki tane 0-kapısının art arda gelmemesi gerektiğinden, devre bir tane 1-kapısına indirgenir ve Şekil 2.20’de gösterilen son halini alır.



Şekil 2. 20. BGM oluşturma 6. Adım

- **Adım 7:** Diyagram nedensellikleri belirlenir (Şekil 2.21).



Şekil 2. 21. BGM oluşturma 7. Adım

Bu devrede, elemanların akım değişkenleri eşit ve gerilim değişkenleri toplamı sıfırdır. Bu veriden yola çıkarak, elemanların 1-kapısı ile birbirine bağlanması gerektiği bu adımlar oluşturulmadan da düşünülebilir. Ancak, karmaşık ve birden fazla enerji modu olan sistemlerde bu tarz çıkarımlar yapmak kolay değildir. Bu yüzden, dinamik sistemlerin BGM'lerinin oluşturulması için bu adımların anlaşılması büyük önem taşımaktadır.

3. FORMÜLASYON VE ÇÖZÜM

3.1. Başlangıç Denklemleri ve Durum-Çıkış Denklemleri

BGM girilen bir sistemin durum-uzay modeline dönüştürülmesi için öncelikle düğümler matrisi oluşturulmalıdır. n_e : BGM’ndeki toplam eleman sayısı ve n_d : BGM’ndeki düğüm sayısı olmak üzere $[n_e \times n_d]$ boyutlu A_b düğümler matrisi,

$$A_b = \left[(a_b)_{i,j} \right]_{n_e \times n_d} \quad (3.1)$$

elde edilir ve elemanları aşağıdaki gibi ifade edilir [51].

$$(a_b)_{i,j} = \begin{cases} 1, j. \text{ çizgi } i. \text{ düğüme bağlı ve seçilmiş akım yönü } i. \text{ düğümden gidiyor ise} \\ -1, j. \text{ çizgi } i. \text{ düğüme bağlı ve seçilmiş akım yönü } i. \text{ düğüme geliyor ise} \\ 0, j. \text{ çizgi } i. \text{ düğüme bağlı değil ise} \end{cases}$$

Referans düğümü çıkarılarak, $[n_e \times n_{d-1}]$ boyutlu A matrisi oluşturulur. Düğüm denklemleri (3.2.a)’daki gibi oluşur. Burada; I : elemanlara ait akımlar vektörüdür, A : bağımsız denklemlerin oluşturduğu düğümler matrisi olup, (3.2.b)’deki gibi iki alt matrise ayrılır.

$$[A] \cdot [I] = [0] \quad (3.2a)$$

$$[A] = [A_1 : A_2] \quad (3.2b)$$

Burada A_1 : bağımsız kolonların oluşturduğu alt matris ve A_2 : diğer kolonların oluşturduğu alt matristir. Temel kesitleme denklemlerini elde etmek için A_1 matrisi, A matrisine elementer satır işlemleri uygulanarak birim matris yapılır. Temel kesitleme denklemleri,

$$[A_1 : A_2] \begin{bmatrix} I' \\ \end{bmatrix} = [0] \quad (3.3)$$

$$\begin{bmatrix} A_1^{-1} A_1 : A_1^{-1} A_2 \end{bmatrix} \begin{bmatrix} I' \end{bmatrix} = [0] \quad (3.4)$$

$$\begin{bmatrix} U_1 : A_1^{-1} A_2 \end{bmatrix} \begin{bmatrix} I' \end{bmatrix} = [0] \quad (3.5)$$

olarak elde edilir. Burada, I' : elemanlara ait değiştirilmiş akımlar vektörü ve U_1 : uygun boyutlu birim matristir. Temel çevre denklemleri de,

$$\begin{bmatrix} -\left(A_1^{-1} A_2\right)^T : U_2 \end{bmatrix} \begin{bmatrix} V' \end{bmatrix} = [0] \quad (3.6)$$

olarak elde edilir. Burada, V' : elemanlara ait gerilimler vektörü ve U_2 : uygun boyutlu birim matristir.

Devredeki kondansatör, indüktans ve cebirsel çok uçlu elemanlara ilişkin tanım bağıntıları, çıkış denklemlerine ilişkin tanım bağıntıları, Kirchoff'un akımlar ve gerilimler kanunu kullanılarak başlangıç denklemleri toplu halde (3.7)'deki gibi yazılır.

$$\begin{bmatrix} C & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e_{18} & e_{25} & e_{32} & e_{39} & e_{46} & 0 & e_{60} \\ 0 & 0 & -A'' & -B'' & -C'' & -D'' & I & -E'' & \\ 0 & 0 & e_{20} & 0 & 0 & e_{41} & e_{48} & 0 & 0 \\ 0 & 0 & 0 & e_{28} & e_{35} & 0 & e_{49} & 0 & e_{63} \end{bmatrix} \cdot \begin{bmatrix} \dot{V}_C \\ \dot{I}_L \\ I_C \\ V_L \\ U_{C\dot{C}U} \\ I_{C\dot{C}U} \\ U_C \\ y \\ U_{SC} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ a_4 & a_{11} \\ A' & B'' \\ 0 & a_{13} \\ a_7 & 0 \end{bmatrix} \cdot \begin{bmatrix} V_C \\ I_L \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ b_4 & b_{11} \\ D' & E' \\ b_6 & b_{13} \\ b_7 & 0 \end{bmatrix} \cdot \begin{bmatrix} U \\ U_{SC} \end{bmatrix} \quad (3.7)$$

Bir sistemin durum-uzay modelinin genel formu;

$$\dot{x}(t) = A \cdot x(t) + B \cdot u(t) \quad (3.8)$$

$$y(t) = C \cdot x(t) + D \cdot u(t) \quad (3.9)$$

şeklinde gösterilmektedir. Denklem (3.8) “Durum Denklemi”, Denklem (3.9) ise “Çıkış Denklemi” olarak adlandırılır. Burada, x : durum değişkenleri vektörü, $\dot{x}$: durum değişkenlerinin türevlerini içeren vektör, y : çıkış vektörü, u : giriş vektörü, A : sistem matrisi, B : giriş matrisi, C : çıkış matrisi ve D : ileri besleme matrisidir.

Sistemin durum-uzay modeli göz önüne alınarak (3.7) daha basit hale indirgenir:

$$\begin{bmatrix} E_{11} & E_{12} & 0 & 0 \\ 0 & E_{22} & 0 & E_{24} \\ 0 & E_{32} & I & E_{34} \\ 0 & E_{42} & 0 & E_{44} \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ Y \\ y \\ U_{SC} \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{A}_1 \\ C_1 \\ \bar{C}_1 \end{bmatrix} \cdot [x] + \begin{bmatrix} 0 & 0 \\ \bar{B}_1 & \bar{B}_1^8 \\ D_1 & D_1^8 \\ \bar{D}_1 & \bar{D}_1^8 \end{bmatrix} \cdot \begin{bmatrix} U \\ U_{SC} \end{bmatrix} \quad (3.10)$$

Buradaki Y vektörü; kapasitör akımları, indüktans gerilimleri ve bağımsız gerilim kaynağının akımı ile cebirsel çok uçlu elemanların gerilimlerinden ve akımlarından oluşan vektördür. Durum ve çıkış denklemlerinin hesaplamalarında Y vektörünün ve çarpanlarının yok edilmesinde bir sakınca bulunmamaktadır. Y vektörünün ve çarpanlarının yok edilmesi işlemi sonucunda başlangıç denklemleri

$$\begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ y \\ U_{SC} \end{bmatrix} = \begin{bmatrix} A \\ C \\ \bar{C} \end{bmatrix} \cdot [x] + \begin{bmatrix} B & B^s \\ D & D^s \\ \bar{D} & \bar{D}^s \end{bmatrix} \cdot \begin{bmatrix} U \\ U_s \end{bmatrix} \quad (3.11)$$

şeklinde olur. Başlangıç denklemleri BONAN programı tarafından elde edilir. Durum ve çıkış denklemleri yukarıda anlatıldığı gibi bulunur.

Devrede çözümü istenen çıkış büyüklükleri elemanların uç değişkenleri olabilir. Bunun yanı sıra uç değişkenlerinin lineer bileşenleri de olabilmektedir. Çözümü bulunması gereken büyüklükler bir y vektöründe toplanırsa y vektörü en genel şekliyle:

$$y = A' \cdot v_C + A'' \cdot i_C + B' \cdot v_L + B'' \cdot i_L + C' \cdot v_{CCU} + C'' \cdot i_{CCU} + D' \cdot u + D'' \cdot u_C + E' \cdot u_s + E'' \cdot u_{SC} \quad (3.12)$$

biçiminde ifade edilebilir.

Programda elde edilen başlangıç denklemlerinden durum değişkenlerinin satırı, Denklem (3.8) ve (3.9) ifadelerini verecek şekilde işleme tabi tutulur. BONAN programı, modele ilişkin (3.8)’de belirtilen A ve B katsayı matrislerini, durum

değişkenlerinin başlangıç değerlerini ve kaynak bilgilerini Matlab ortamında çözdürülmek üzere .m uzantılı bir dosyaya aktarır. Elde edilen durum ve çıkış denklemlerine, tam çözüm formülasyonu uygulanır.

Durum denklemlerinin tam çözümleri incelenirken;

$$x_{(tam)}(t) = x_{(öz)}(t) + x_{(zor)}(t) \quad (3.13)$$

çözüm ifadelerinde “öz çözüm” ve “zorlanmış çözüm” terimleri bulunmaktadır. Öz çözüm, kaynaklar devre dışı edildikten sonra kondansatör geriliminin başlangıç değeri ve indüktans akımının başlangıç değeri ile devrenin çözümüdür. Zorlanmış çözüm ise, kaynaklar devredeyken ve başlangıç değerleri sıfır alınarak yapılan çözümdür. Tam çözüm, öz çözüm ve zorlanmış çözüm ifadelerinin toplamıdır.

Öz çözüm, devredeki tüm bağımsız kaynakların devre dışı bırakılmasıyla ve sadece başlangıç şartlarının değerlendirilmesi ile bulunur. Denklem (3.8)’de bağımsız kaynaklar devre dışı edilirse:

$$\dot{x}(t) - A \cdot x(t) = 0 \quad (3.14)$$

şeklinde homojen denklem takımı oluşur. Bu denklem takımının homojen genel çözümü yapılır ve başlangıç şartları yerine yazılırsa öz çözüm,

$$X_{öz}(t) = X(0) \cdot e^{A \cdot t} \quad (A \in R^{n \times n}) \quad (3.15)$$

olur. Burada, $e^{A \cdot t}$: durum geçiş matrisi ve n: durum değişkeni sayısıdır.

Zorlanmış çözüm, bütün başlangıç koşulları sıfır olduğu durumda sadece bağımsız kaynakların etkisi ile ortaya çıkan çözümdür. Homojen genel çözüm ile özel çözümün toplamına eşittir:

$$X_{zor}(t) = X_{hg}(t) + X_{özel}(t) \quad (3.16)$$

Matlab ortamında devrelerin tam çözümü için, sırasıyla öz ve zorlanmış çözümlerin s-domenin karşılıkları bulunup t-domenine dönüştürülür. Denklem (3.8)'de her iki tarafın Laplace dönüşümü alınır;

$$s \cdot X(s) - X(0) = A \cdot X(s) + B \cdot U(s) \quad (3.17)$$

$$sI \cdot X(s) - A \cdot X(s) = X(0) + B \cdot U(s) \quad (3.18)$$

$$(sI - A) \cdot X(s) = X(0) + B \cdot U(s) \quad (3.19)$$

$$X(s) = (sI - A)^{-1} \cdot X(0) + (sI - A)^{-1} \cdot B \cdot U(s) \quad (3.20)$$

elde edilir. Burada öz çözüm;

$$X_{(öz)}(s) = (sI - A)^{-1} \cdot X(0) \quad (3.21)$$

olarak ifade edilir. Zorlanmış çözüm ise;

$$X_{(zor)}(s) = (sI - A)^{-1} \cdot B \cdot U(s) \quad (3.22)$$

şeklinde s-domeninde yazılabilir. Her iki ifadenin de Ters Laplace Dönüşümü alınırsa öz ve zorlanmış çözümler ayrı ayrı bulunabilir.

4. BONAN PROGRAMI

4.1. Giriş

Bilgisayar sistemleri yazılım ve donanım olarak iki kısımdan oluşur. Yazılım, bilgisayar donanımını çeşitli elektriksel sinyallerle kontrol etmeye yarayan komutlar bütünüdür. Modern programlama dillerinin ortaya çıkışı, ilk elektrikli bilgisayarın yapıldığı 1940 yılına dayanmaktadır. İlk bilgisayarlar sınırlı bir hafızaya ve işlem gücüne sahip olduğu için Assembly dili bilgisayarların programlanmasında yeterliydi. Ancak gelişen teknolojiyle birlikte artan ihtiyaçlar, bu dilde kod yazmayı zorlaştırıyordu ve Assembly dilinin hatalara açık bir dil olduğu ortaya çıktı. İlerleyen zamanlarda; Simula(1962), BASIC(1964), Pascal(1970), C(1972), Python(1991), Java(1991) gibi birçok programlama dili geliştirilmiştir [49].

Java, internet teknolojisinin gelişmesiyle ortaya çıkmış ve platform bağımsızlığını amaç edinen bir programlama dilidir. Açık kaynak kodlu olması, ortaya çıkışından günümüze kadar büyük bir kullanıcı topluluğu tarafından tercih edilmiştir. Günümüzde çok çeşitli alanlarda kullanılan ve yazılan kodları nerdeyse hiçbir değişikliğe gitmeden, tüm işletim sistemlerinde kullanabilir olan nesne tabanlı bir dildir. Platform bağımsızlığı, güçlü bir kullanıcı topluluğu desteği, bedava olmasının yanı sıra Netbeans, Eclipse, Jdeveloper gibi gelişmiş ve çoğu açık kaynak kodlu Entegre Geliştirme Ortamları'na (Integrated Development Environment) sahiptir[49].

Sınıflar ve nesneler, Java programlama dilinin temelini oluşturmaktadır. Nesne tabanlı bir programda, sınıf (class) kavramı iyi bilinmelidir. Sınıflar, sahip oldukları değişkenler sayesinde anlık durum bilgilerini tutabilirler. Bunun yanı sıra, sınıflar içinde üye değişkenlerine erişimi sağlayarak sınıfların davranışını değiştiren ya da sahip oldukları durum hakkında bir çıkış vererek dış dünyayı bilgilendiren çeşitli metotlar da tanımlanabilir. Nesneler ise sınıflarda tanımlı olan şablonlara uygun örneklemelerdir. Örneğin bitki kelimesi bir sınıf; yasemin, hanımeli, orkide ve menekşe ise bitki sınıfının nesneleri olarak düşünülebilir.

Netbeans, Java programlama dilinin geliştirilmesinde güçlü bir ortam olarak adından söz ettirmektedir. Açık kaynak kodlu olması, Java ile ilgili tüm uygulama

yapılarına entegre bir şekilde destek sunması, masaüstü uygulamalar için kullanışlı bir arayüze sahip olması, çabuk öğrenilmesi ve çok güçlü bir Java editörüne sahip olması ile Netbeans bu tez çalışmasında geliştirme ortamı olarak seçilmiştir [49].

Geliştirilen program, Java dili ve Netbeans ortamında oluşturulmuştur. Program, bir masaüstü uygulaması olarak Andreas Tetzl tarafından yazılan, lojik işlemler yapan açık kaynak kodlu LogicSim programının tezin amacına uygun olarak değiştirilmesi ile yeniden düzenlenmiştir. Şekil 4.1, LogicSim hakkında bilgiler içermektedir [50].



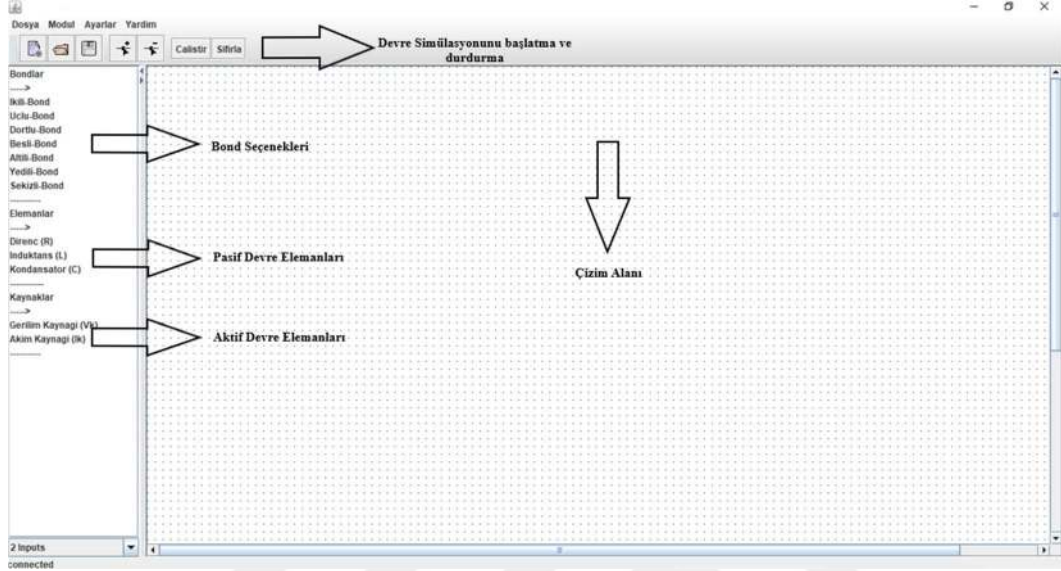
Şekil 4. 1. LogicSim hakkında genel bilgiler

LogicSim, çeşitli lojik kapılar ile dijital lojik devre simülasyonunun yapıldığı bir programdır. LogicSim, Java programlama dili ile yazılmış olup arayüz oluşturma algoritmasının temel özellikler taşıması, kodlama yapısının anlaşılabilirliği ve yapılacak değişimlere açık olması gibi faydalar içerdiğinden arayüz oluşturma sürecinin kısaltılması amacıyla bu çalışmada tercih edilmiştir. Amacına uygun olarak tüm menüleri ve nesne özellikleri yeniden tasarlanarak kodlanmıştır. Görsel olarak LogicSim program yapısına büyük oranda bağlı kalınmasına rağmen işlemsel özellikleri tümüyle farklıdır.

4.2. BONAN Programının Yapısı

BONAN programı masaüstü uygulaması olarak kullanılmaktadır. Programın başlangıç sayfası Şekil 4.2.'de gösterilmektedir. Sayfanın sol kısmında bond menüsü, devre elemanları menüsü ve kaynak menüsü bulunmaktadır. Sağ üst kısımda ise

devrenin simülasyonunun başlatılması ve tamamlanması için “Çalıştır” ve “Sıfırla” butonları bulunmaktadır. Devre çizimi için orta alanda ızgaralı bir bölüm oluşturulmuştur.



Şekil 4. 2. BONAN program açılış sayfası

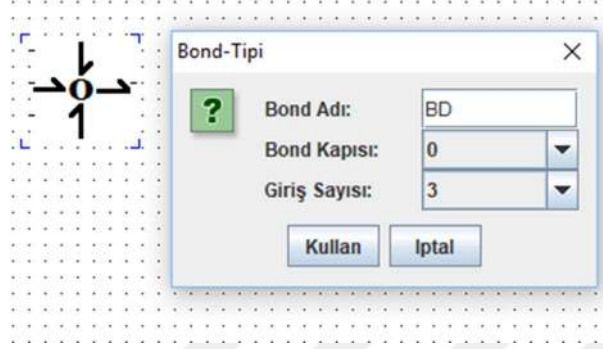
4.2.1. Bond Elemanları Menüsü

Bond elemanları menüsü, ikili bond ila sekizli bond arasında 7 çeşit bond seçeneğinden oluşur. Bond elemanları menüsü, Şekil 4.3.’de gösterilmiştir. Bond çeşitleri, her bir 0 veya 1- kapısına bağlı eleman sayılarına göre oluşturulmuştur. Örneğin, devrede 1- kapısına bağlı 3 eleman varsa bond elemanları menüsünden “Uclu-Bond” seçilmelidir.



Şekil 4. 3. Bond elemanları menüsü

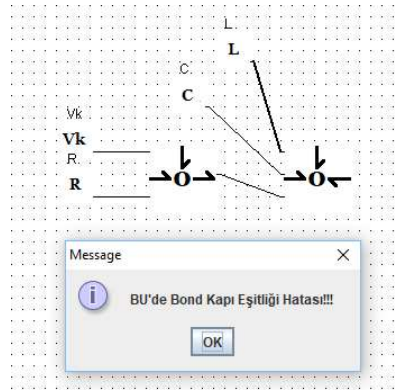
Bond elemanlarını seçerek model oluşturmak için uygun bond sürüklenerek çizim alanına bırakılır. Daha sonra bond tipini, adını ve giriş sayısını girebilmek için bonda sağ tıklanarak “Özellikler” kısmı açılır. Bu kısım Şekil 4.4.’de gösterilmiştir.



Şekil 4. 4. Bond elemanlarına ait özellikler kısmı

Bond isimleri, bondun giriş ve çıkış sayılarının toplamına göre isimlendirilir. Örneğin; 3 girişe ve 1 çıkışa sahip bir bond kullanılacaksa menüden “Dortlu Bond” elemanı seçilir. Bu eleman, **Bond-Dortlu** isminin baş harflerinden yola çıkılarak “BD” olarak adlandırılmıştır. Devrede birden fazla “Dortlu Bond” kullanılacaksa, bondlar “BD1”, “BD2”,..., “BDn” olarak isimlendirilir.

Bond tipi, bondun 0 veya 1- kapısı olmasını tanımlar. “Özellikler” kısmında ‘Bond Kapısı’ seçeneği modele uygun olarak 0 veya 1- kapısı olarak seçilmelidir. Bond tipi seçilirken, iki tane 0- kapısının veya iki tane 1- kapısının art arda gelmemesine dikkat edilmelidir. Bu durum oluşursa program, Şekil 4.5’de görüldüğü gibi “Bond Kapı Eşitliği” hatası verir ve model analiz edilemez.



Şekil 4. 5. Bond kapı eşitliği hatasının gösterimi

Bond elemanlarının giriş sayısı, bondun başka bir bonda bağlı olup olmamasına göre değişir. Çıkışında başka bir bonda bağlantısı bulunan bondların giriş sayısı, toplam sayının bir eksiklidir. Örneğin, girişinde 3 eleman bulunan ve çıkışı da başka bonda bağlı olan bond, “Dortlu-Bond” seçilmeli ve giriş sayısı 3 olmalıdır. Ama çıkışına bir bond bağlı değilse o zaman “Dortlu-Bond” yerine “Uclu-Bond” seçilerek giriş sayısı 3 yapılmalıdır.

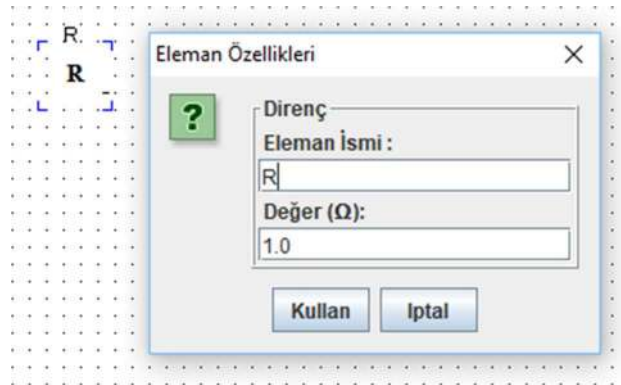
4.2.2. Pasif Devre Elemanları Menüsü

Pasif devre elemanları menüsünde direnç (R), kondansatör (C) ve indüktans (L) elemanları bulunmaktadır. Pasif elemanlar menüsünün görsel kullanıcı arayüzündeki gösterimi, Şekil 4.6.’da verilmiştir.



Şekil 4. 6. Pasif devre elemanları menüsünün gösterimi

Direnç elemanının “Özellikler” kısmı seçildiğinde, Şekil 4.7.’de gösterilen ekrana elemanın ismi ve değeri girilmelidir. İki veya daha fazla direnç elemanını içeren bir model oluşturulurken, herbir direnç elemanına ilişkin eleman isimleri farklı olmak zorundadır. Elemanın ismi değiştirildiğinde, bu değişim direnç görselinin üstünde güncellenecek şekilde tasarım yapılmıştır.

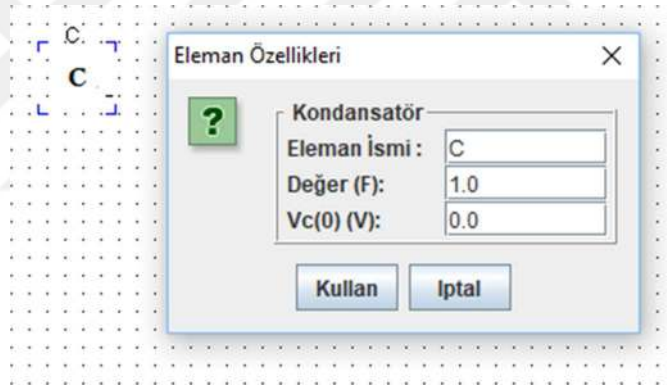


Şekil 4. 7. Direnç elemanının özellikler kısmı

Direnç elemanından farklı olarak, indüktans ve kondansatör elemanlarının özellikler kısmında elemanların başlangıç değerlerine de yer verilmiştir. İndüktans elemanına ilişkin başlangıç akımı ve kondansatör elemanına ilişkin başlangıç gerilimi, eleman ismine ve değerine ek olarak girilebilmektedir. İndüktans elemanının “Özellikler” kısmı Şekil 4.8.’de, kondansatör elemanınki ise Şekil 4.9.’da gösterilmiştir.



Şekil 4. 8. İndüktans elemanının özellikler kısmı



Şekil 4. 9. Kondansatör elemanının özellikler kısmı

4.2.3. Aktif Devre Elemanları Menüsü

Aktif devre elemanları menüsünde, elektrik devrelerine ilişkin bağımsız gerilim ve akım kaynakları bulunmaktadır. Kaynakları içeren menünün gösterimi, Şekil 4.10.’da verilmiştir.



Şekil 4. 10. Kaynaklar menüsünün gösterimi

Bağımsız akım ve gerilim kaynaklarının, isimleri, genlik değerleri, kaynak tipi ve alternatif akım ya da gerilim kaynağı olması durumunda açısal frekans ve kaynak açısı değerleri girilebilmektedir. Gerilim ve akım kaynaklarına ait özellikler Şekil 4.11. ve Şekil 4.12.’de gösterilmektedir. Kaynaklar “DC” olarak seçildiğinde açısal frekans ve kaynak açısı değerleri program tarafından 0 alınır. “AC” kaynak seçiminde (4.1) ve (4.2)’de belirtilen açısal frekans ve kaynak açısı değerleri girilmek zorundadır.

$$V_k(t) = V_m \cdot \sin(\omega \cdot t + \theta_v) \quad (4.1)$$

$$I_k(t) = I_m \cdot \sin(\omega \cdot t + \theta_i) \quad (4.2)$$

Şekil 4. 11. Gerilim kaynağının özellikler kısmı

Şekil 4. 12. Akım kaynağının özellikler kısmı

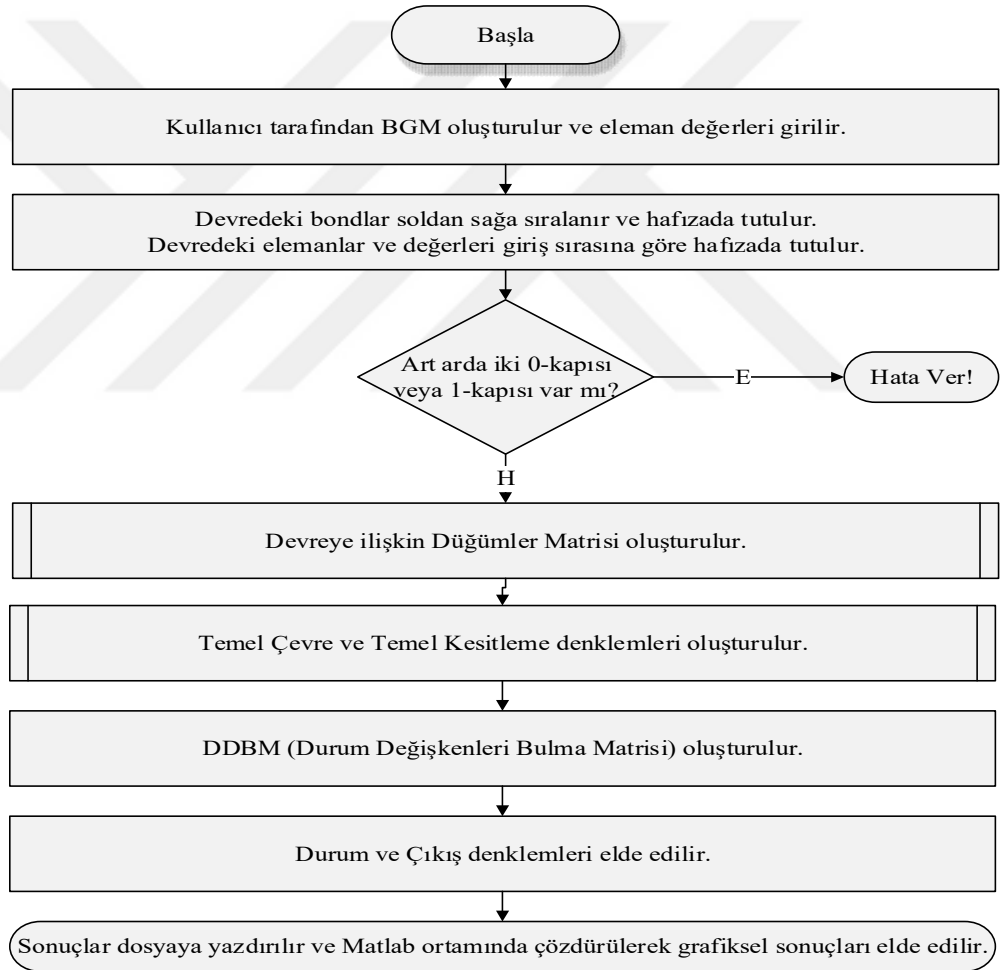
Devre çizimi için menülerden seçilen uygun elemanlar, üzerlerine tıklanıp sürüklenerek çizim alanına bırakılır. Bütün elemanların değerleri, başlangıçta 1 olarak atanmaktadır. Elemanlar farklı bir değere sahip ise çizim alanında üzerine sağ ile tıklanarak açılan menülerden “Özellikler” kısmında değeri girilir. Elemanlar bondlara bağlandıktan sonra üst menüde bulunan “Çalıştır” ve “Sıfırla” butonlarına sırasıyla basılır ve ekrana gelecek katsayı matrisleri Matlab ortamında çözümün yapılacağı dosyaya aktarılır.

Programın hatasız çalışması için aşağıda belirtilen durumlara dikkat edilmesi gerekir:

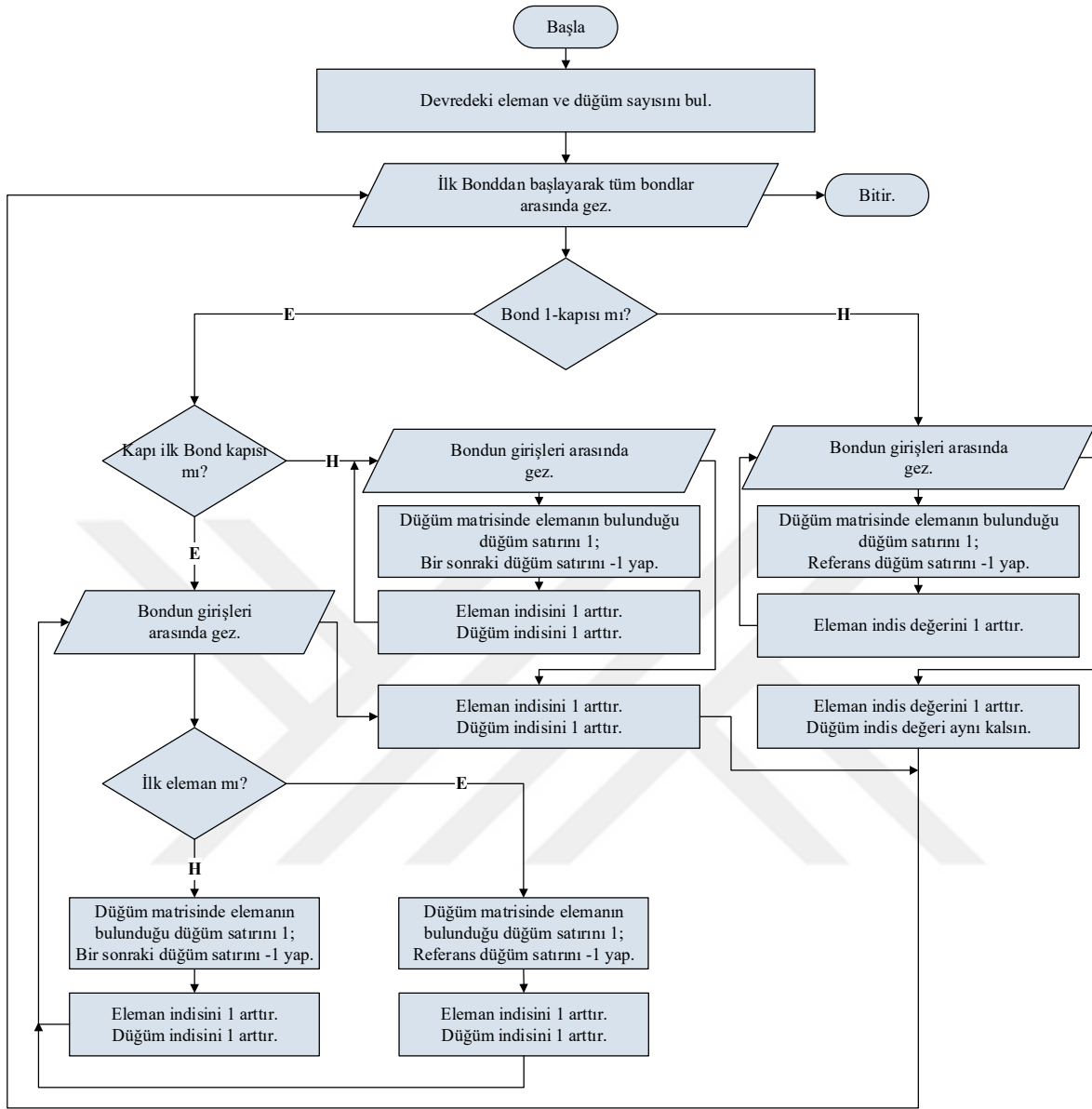
- Bondlar, devredeki sıralarına uygun olarak soldan başlanarak çizim alanına alınmalıdır.
- Eleman isimleri, birbirlerinden farklı olmak zorundadır.
- İlk bondda kaynak bulunuyorsa, ilk girişe kaynak bağlanmalıdır.
- Bir bond diğerine bağlanırken; ilk bondun çıkışı ikinci bondun en son girişine denk gelecek biçimde bağlantı yapılmalıdır.
- İki tane 1-kapısı veya iki tane 0-kapısı art arda gelmeyecek biçimde bağlantı yapılmalıdır. Aksi takdirde, bu durum BG yönteminin doğasına aykırı olacağından program Şekil 4.5.’ gösterildiği gibi “Bond Kapı Eşitliği” hatası verecektir.
- Kaynak “AC” ise “Açısal Frekans” değeri 0’dan küçük olursa, program hata verir.

4.3. BONAN Programı Akış Şeması

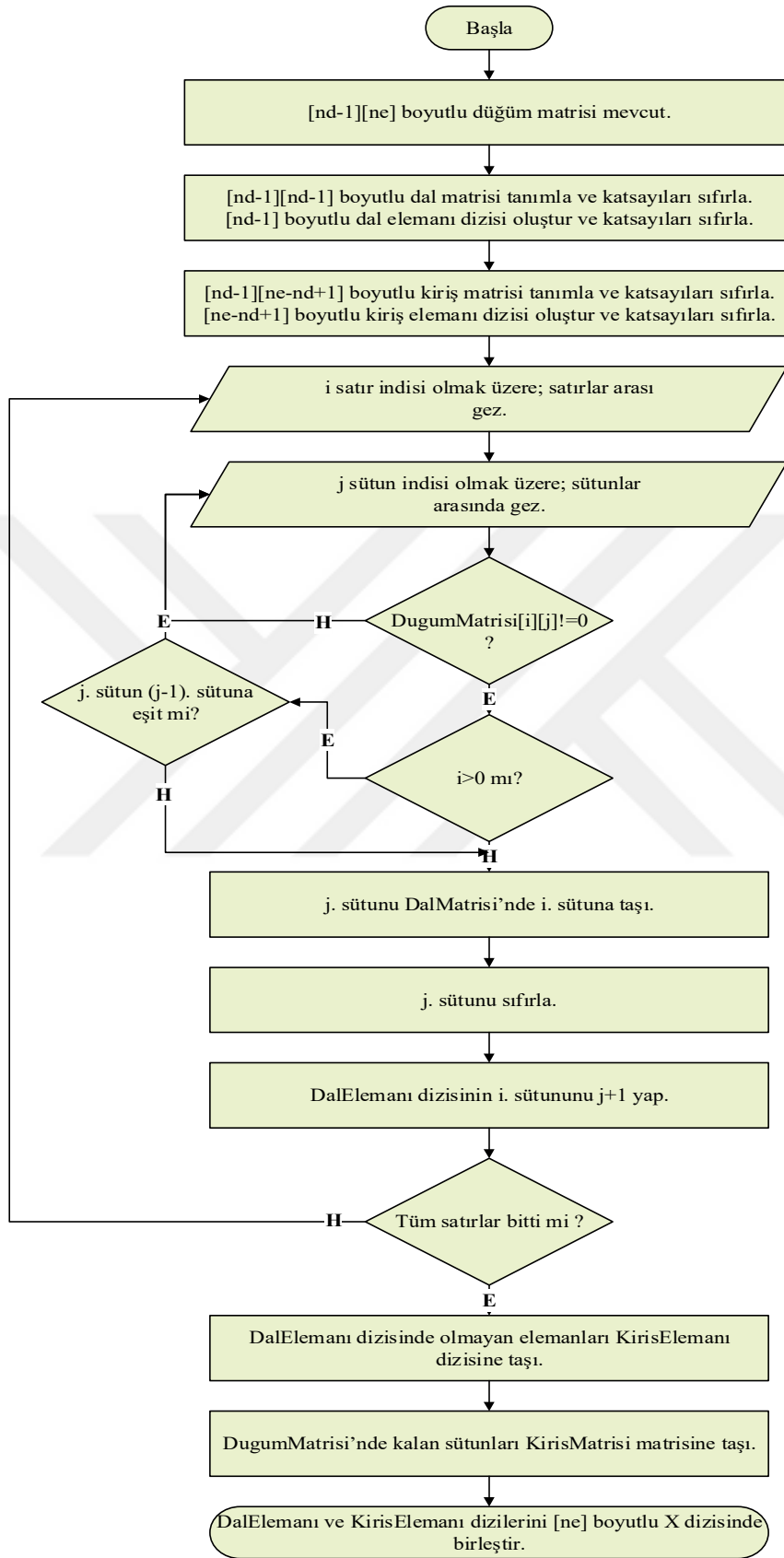
Tez kapsamında oluşturulan BONAN programına ilişkin genel akış diyagramı, Şekil 4.13.'de verilmiştir. Bu akış diyagramı içerisinde belirtilen düğüm matrisi oluşturma alt diyagramı Şekil 4.14.'de; düğüm matrisi içerisinde dal ve giriş elemanlarını seçme alt diyagramı Şekil 4.15.'de; dal ve giriş katsayı matrislerinden temel çevre ve temel kesitleme matrislerini elde etme alt diyagramı Şekil 4.16.'da verilmiştir.



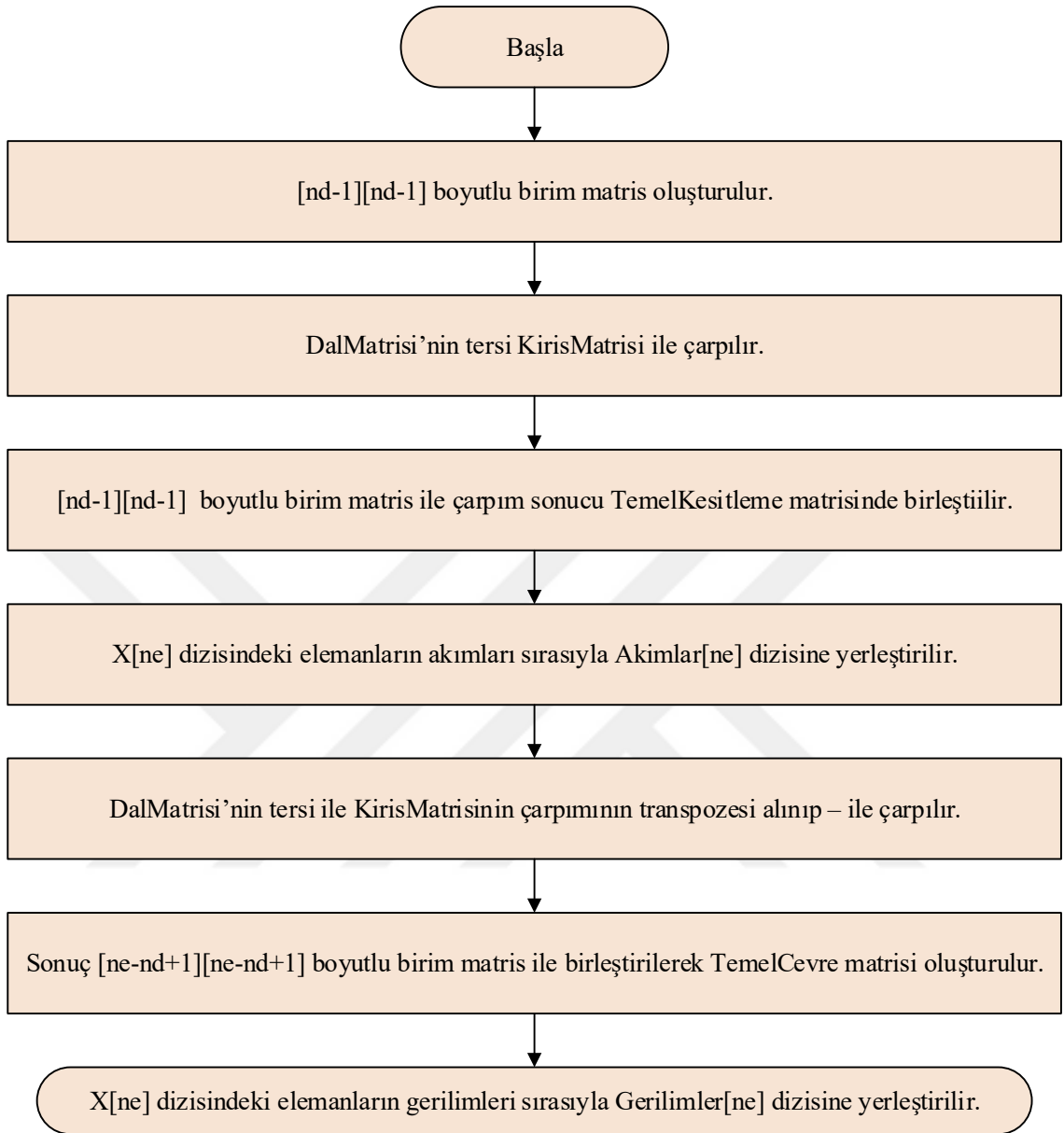
Şekil 4. 13. BONAN programının genel akış diyagramı



Şekil 4. 14. Düğüm matrisi oluşturma alt akış diyagramı



Şekil 4. 15. Düğüm matrisi içerisinde dal ve kiriş elemanlarını seçme alt diyagramı

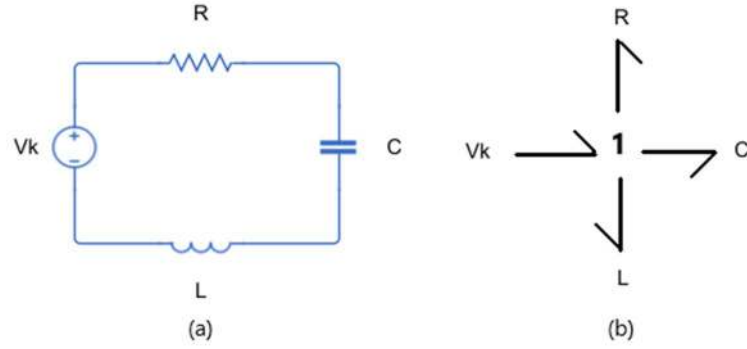


Şekil 4. 16. Dal ve giriş katsayı matrislerinden, temel çevre ve temel kesitleme matrislerini elde etme alt diyagramı

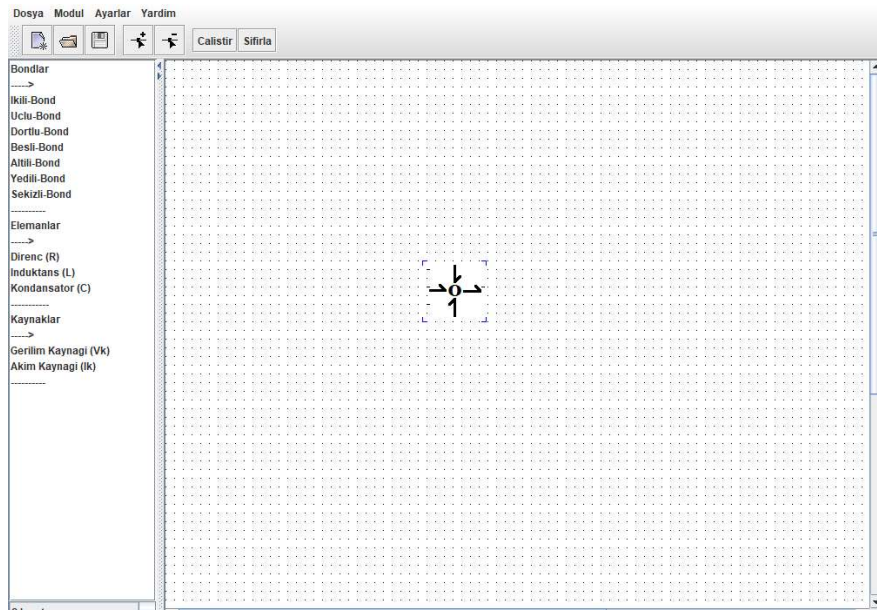
5. ÖRNEK DEVRELER

5.1. Örnek 1

Bu örnekte incelenecek devre, Şekil 5.1.a'da verilmiştir. Şekil 5.1.b'de görülen BG modelini oluşturmak için, programın ana ekranında sol kısımda bulunan menüden uygun elemanlar sırasıyla ızgaralı devre çizim alanına yerleştirilir. Örnek devrenin 4 elemandan ve bir temel çevreden oluştuğu göz önüne alınırsa, Şekil 5.2.'deki gibi 4 kapılı "1" bondun kullanılması gerekmektedir.



Şekil 5. 1. Örnek 1'e ilişkin (a) devre şeması ve (b) BG modeli



Şekil 5. 2. Örnek 1'e ilişkin uygun bond seçimi

Bondu çizim alanına yerleştirdikten sonra eleman özelliklerini girmek amacıyla elemana sağ tıklanarak Şekil 5.3.'de belirtildiği gibi elemanın özellikler kısmı açılmalıdır.



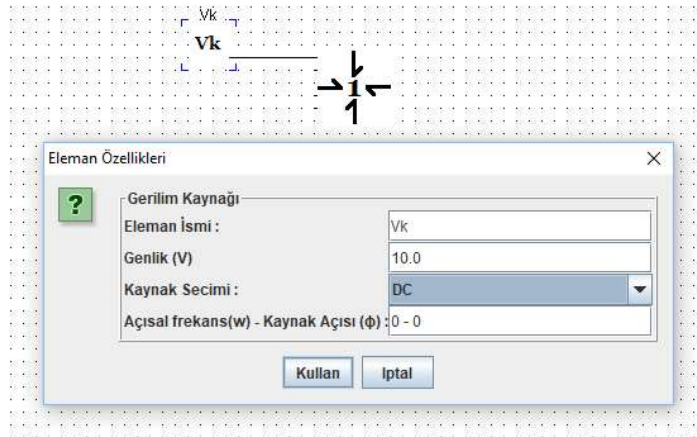
Şekil 5. 3. Elemanlara ilişkin özellikleri seçme menüsü

Elemana ilişkin özellikler kısmından 4 girişli “1” kapısı için Şekil 5.4.’deki gibi veri girişi yapılır. Açılan bu pencerede bonlara ilişkin bond adı, kapı tipi ve giriş sayısı değerleri değiştirilebilir.



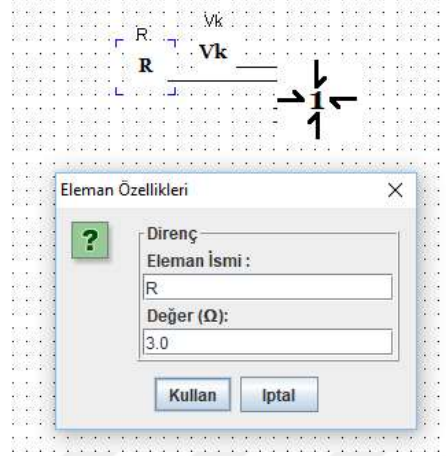
Şekil 5. 4. Örnek 1'e ilişkin bond özellikleri

V_k gerilim kaynağının özellikleri, Şekil 5.5.’de gösterildiği gibi seçilir. Kaynak özellikleri kısmında sırasıyla kaynağın ismi, değeri, tipi ve açısal frekans, kaynak açısı değişkenleri bulunmaktadır. Örnek devrede kaynak “DC” özellikte olduğundan, açısal frekans ve kaynak açısı değerleri boş bırakılır.

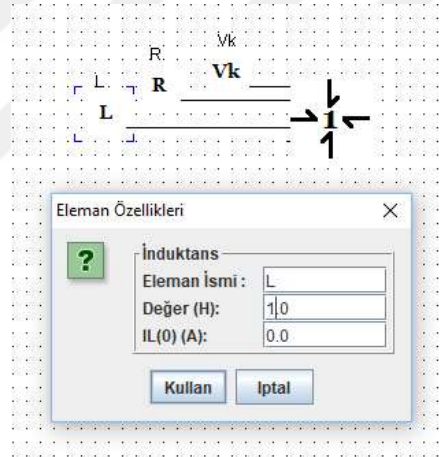


Şekil 5. 5. Örnek 1'e ilişkin kaynak verileri

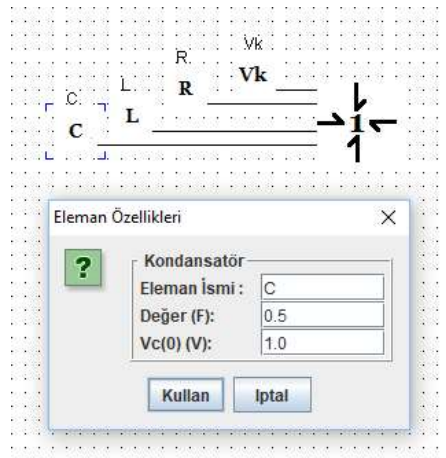
Kaynak eklendikten sonra sırasıyla direnç, indüktans ve kondansatör elemanları değerleri ile birlikte girilerek bonda eklenir. Elemanların değerleri Şekil 5. 6., Şekil 5. 7. ve Şekil 5. 8.'de belirtilmiştir.



Şekil 5. 6. Örnek 1'e ilişkin direnç verileri

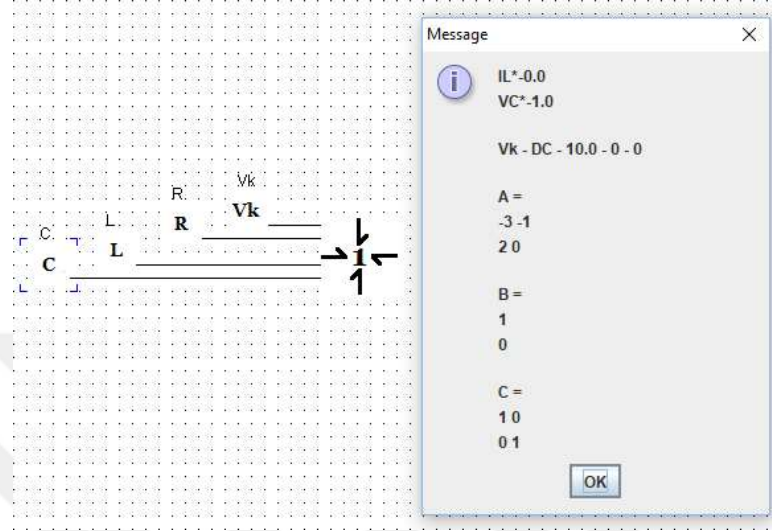


Şekil 5. 7. Örnek 1'e ilişkin indüktans verileri



Şekil 5. 8. Örnek 1'e ilişkin kondansatör verileri

Devre modeli ve veri giriři tamamlandıktan sonra, üst menüde bulunan “Çalıştır” ve “Sıfırla” butonlarına sırasıyla basılır. Program ilgili katsayı matrisleri ile durum deęişkenleri ve kaynak bilgilerini ekranda gösterir. Şekil 5.9.’ da devrenin program çıktıları ile birlikte son hali verilmiştir.



Şekil 5. 9. Örnek 1'e ilişkin BONAN program çıktıları

Böylece program tarafından durum ve çıkış denklemleri,

$$\begin{bmatrix} \dot{I}_L \\ \dot{V}_C \end{bmatrix} = \begin{bmatrix} -3 & -1 \\ 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} I_L \\ V_C \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot V_k \quad (5.1)$$

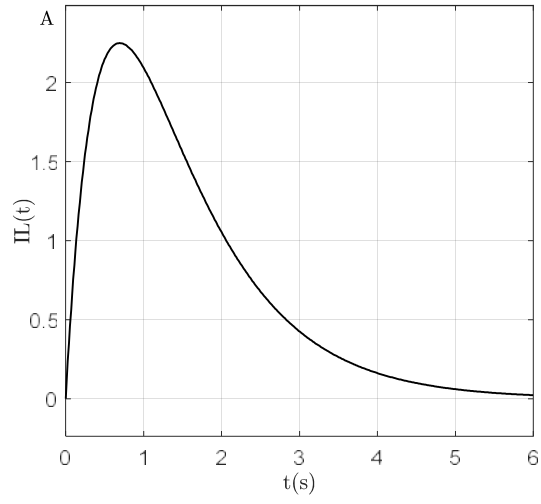
$$\begin{bmatrix} I_L \\ V_C \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} I_L \\ V_C \end{bmatrix} \quad (5.2)$$

şeklinde elde edilir. Program çıktıları belirtilen herhangi bir .m uzantılı bir dosyaya aktarılır. Matlab ortamında bu dosya açılır, katsayı matrisleri ile başlangıç şartları ve kaynak bilgileri kullanılarak devrenin tam çözümü yapılır. \$I_L\$ ve \$V_C\$'ye ilişkin sonuçların zamana göre ifadeleri,

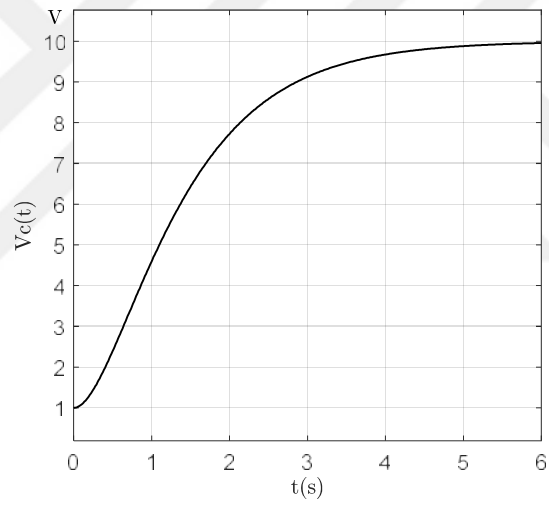
$$I_{L(tam)}(t) = 9 \cdot e^{-t} - 9 \cdot e^{-2t} \quad (5.3)$$

$$V_{C(tam)}(t) = -18 \cdot e^{-t} + 9 \cdot e^{-2t} + 10 \quad (5.4)$$

olur. Bu büyüklüklerin zamana göre deęişimleri, Şekil 5.10 ve 11'de verilmiştir.



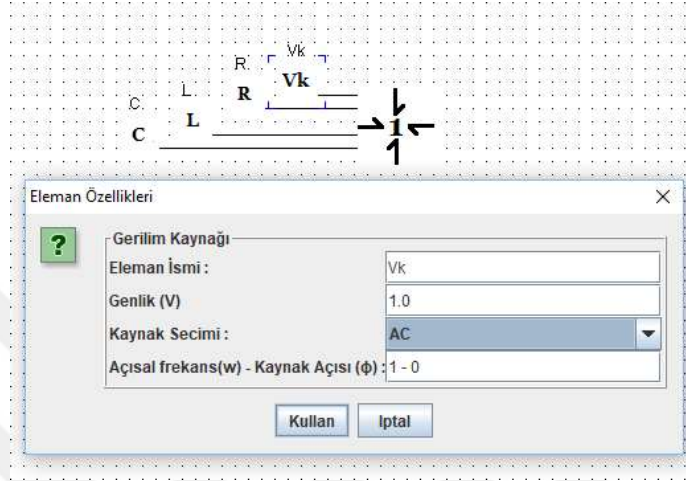
Şekil 5. 10. Örnek 1'e ilişkin indüktans akımının değişimi



Şekil 5. 11. Örnek 1'e ilişkin kondansatör gerilimi değişimi

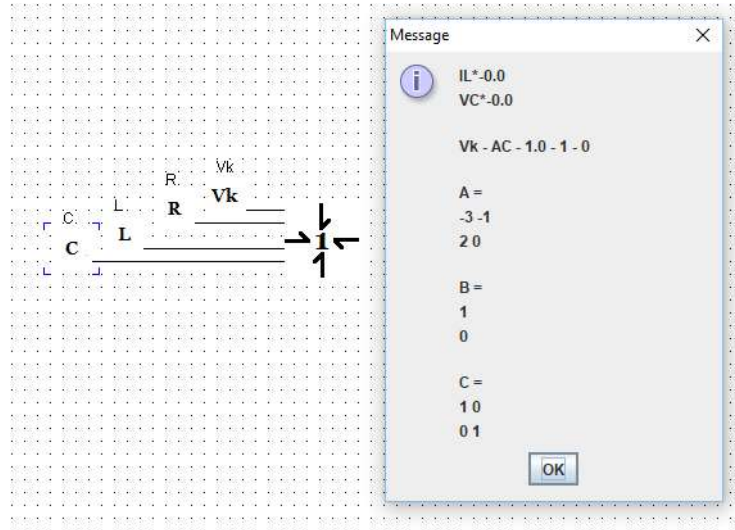
5.2. Örnek 2

Bu örnekte, örnek 1'deki devrenin kaynağı $V_k = \sin(t)$ olarak alınmıştır ve . Bu nedenle kaynak tipi kısmı “AC” şeklinde değiştirilmiştir ve açısal frekans ile kaynak açısı değerleri uygun şekilde girilmiştir (Şekil 5. 12).



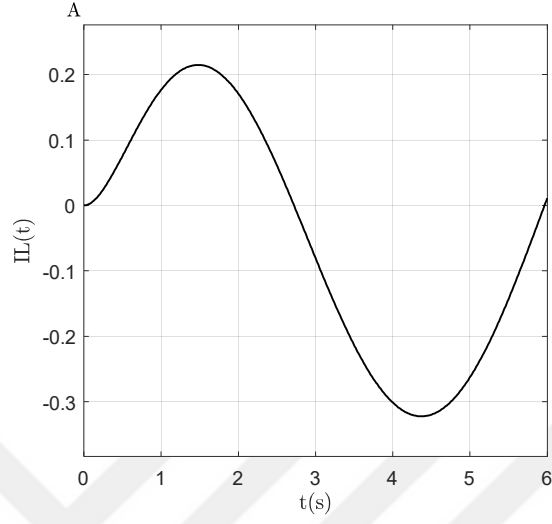
Şekil 5. 12. Örnek 2'ye ilişkin kaynak verileri

Kaynak uygun şekilde ayarlanıp program çalıştırıldığında Şekil 5. 13.'de bulunan katsayı matrisleri ve durum değişkeni ile kaynak değerleri ekrana bildirim mesajı olarak gelir.

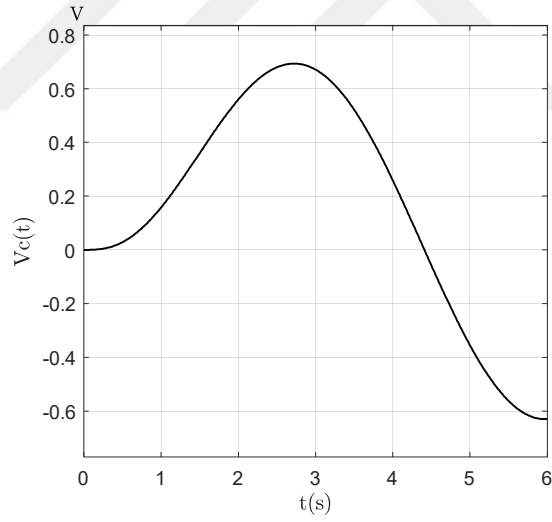


Şekil 5. 13. Örnek 2'e ilişkin BONAN program çıktıları

Bu deęerler Matlab ortamında özdürüldüğünde Şekil 5. 14 ve 5.15’de görölen program ıktılarını verir.



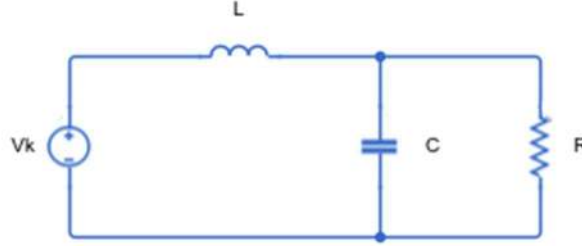
Şekil 5. 14. Örnek 2'e ilişkin indüktans akımının deęiřimi



Şekil 5. 15. Örnek 2'e ilişkin kondansatör gerilimi deęiřimi

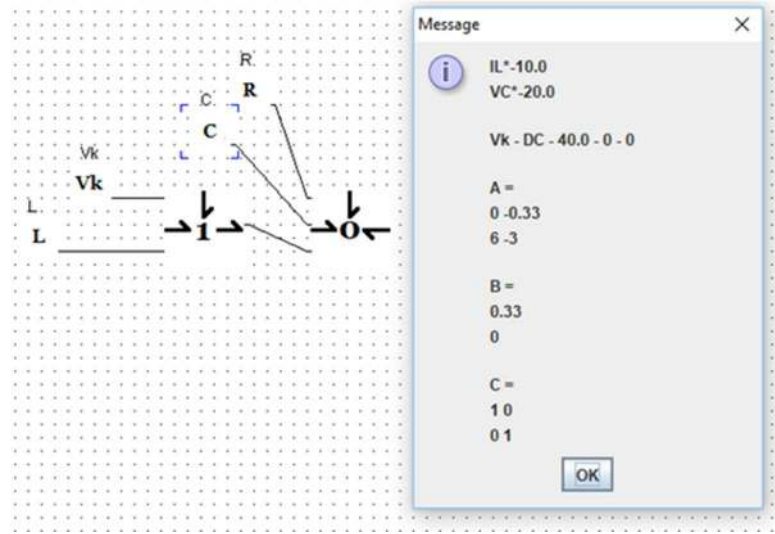
5.3. Örnek 3

Bu örnekte Şekil 5.16.'da verilen devre kullanılmıştır. DC kaynak kullanılmış ve genliği 40V olarak seçilmiştir. Eleman değerleri $L=3H$, $R=2\Omega$, $C=1/6F$ olup, başlangıç değerleri $I_L(0) = 10A$ ve $V_C(0) = 20V$ olarak belirlenmiştir.



Şekil 5. 16. Örnek 3'e ilişkin devre şeması

Devrenin BGM ve program çıktıları Şekil 5. 17.'da görünmektedir.



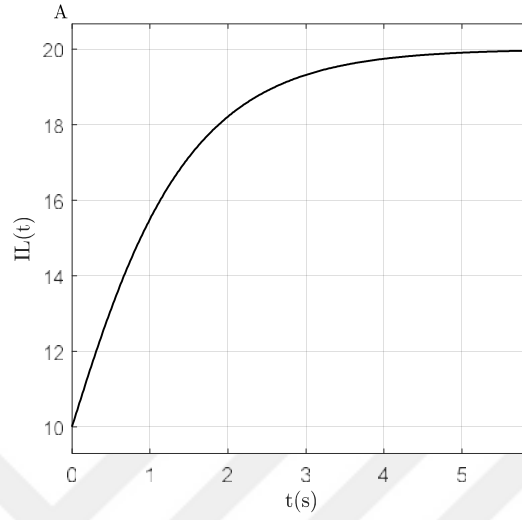
Şekil 5. 17. Örnek 3'e ilişkin BONAN program çıktıları

Program tarafından, (5.5) ve (5.6)'de verilen durum ve çıkış denklemleri elde edilir.

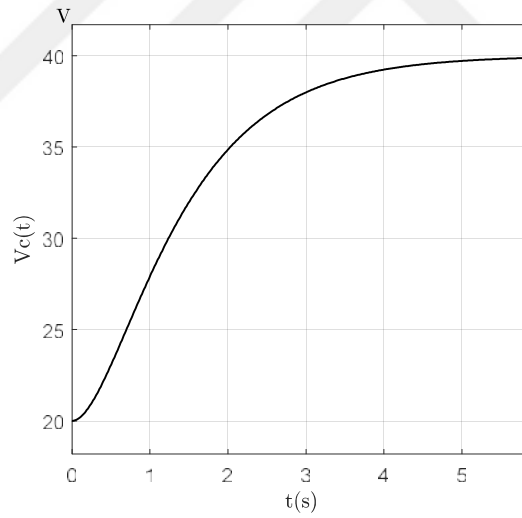
$$\begin{bmatrix} \dot{I}_L \\ \dot{V}_C \end{bmatrix} = \begin{bmatrix} 0 & -0.33 \\ 6 & -3 \end{bmatrix} \cdot \begin{bmatrix} I_L \\ V_C \end{bmatrix} + \begin{bmatrix} 0.33 \\ 0 \end{bmatrix} \cdot V_k \quad (5.5)$$

$$\begin{bmatrix} I_L \\ V_C \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} I_L \\ V_C \end{bmatrix} \quad (5.6)$$

Örnek 3'e ilişkin durum değişkenlerinin değişim grafikleri Şekil 5. 18. ve 5.19'da verilmiştir.



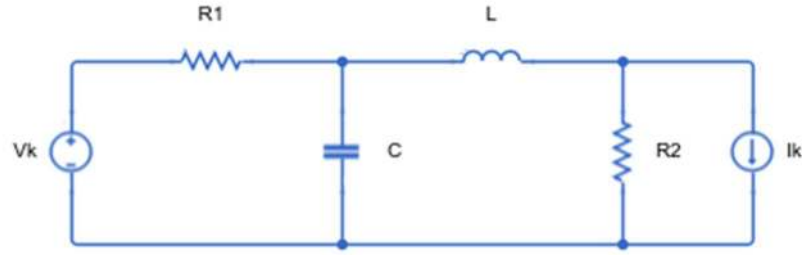
Şekil 5. 18. Örnek 3'e ilişkin indüktans akımı değişimi



Şekil 5. 19. Örnek 3'e ilişkin kondansatör gerilimi değişimi

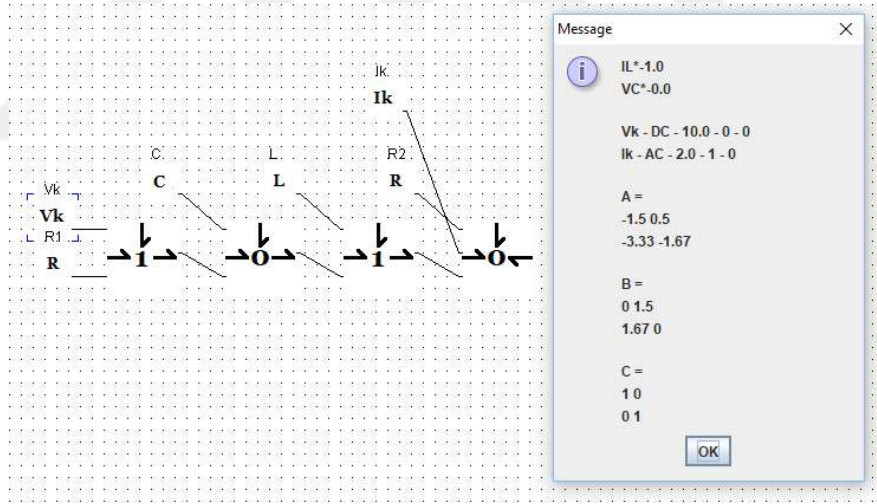
5.4. Örnek 4

Bu örnekte incelenen Şekil 5.20.'de gösterilen devrede, bir DC bir de AC olmak üzere iki tane kaynak bulunmaktadır. Eleman değerleri $R_1=2\Omega$, $R_2=3\Omega$, $C=0.3F$, $L=2H$; kaynak değerleri $V_k=10V$, $I_k=2\sin(t)$; başlangıç değerleri $I_L(0) = 1A$ ve $V_C(0) = 0V$ olarak belirlenmiştir.



Şekil 5. 20. Örnek 4'e ilişkin devre şeması

Devrenin BGM ve program çıktıları Şekil 5. 21.'de görünmektedir.



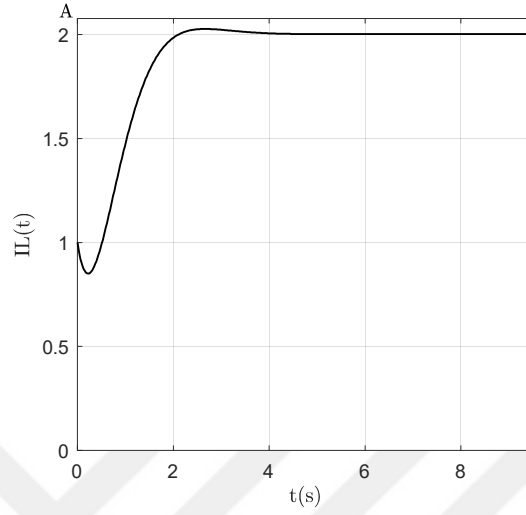
Şekil 5. 21. Örnek 4'e ilişkin BONAN program çıktıları

Program tarafından, (5.7) ve (5.8)'de verilen durum ve çıkış denklemleri elde edilir.

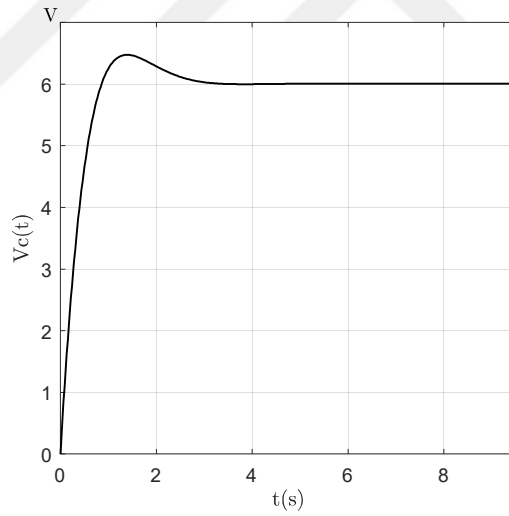
$$\begin{bmatrix} \dot{I}_L \\ \dot{V}_C \end{bmatrix} = \begin{bmatrix} -1.5 & 0.5 \\ -3.33 & -1.67 \end{bmatrix} \cdot \begin{bmatrix} I_L \\ V_C \end{bmatrix} + \begin{bmatrix} 0 & 1.5 \\ 1.67 & 0 \end{bmatrix} \cdot \begin{bmatrix} V_k \\ I_k \end{bmatrix} \quad (5.7)$$

$$\begin{bmatrix} I_L \\ V_C \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} I_L \\ V_C \end{bmatrix} \quad (5.8)$$

Örnek 4'e ilişkin durum değişkenlerinin değişim grafikleri Şekil 5. 22. ve 5.23'de verilmiştir.



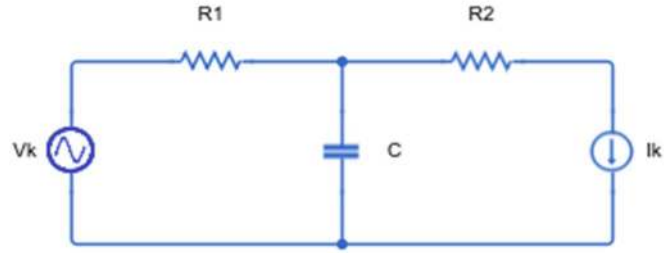
Şekil 5. 22. Örnek 4'e ilişkin indüktans akımı değişimi



Şekil 5. 23. Örnek 4'e ilişkin kondansatör gerilimi değişimi

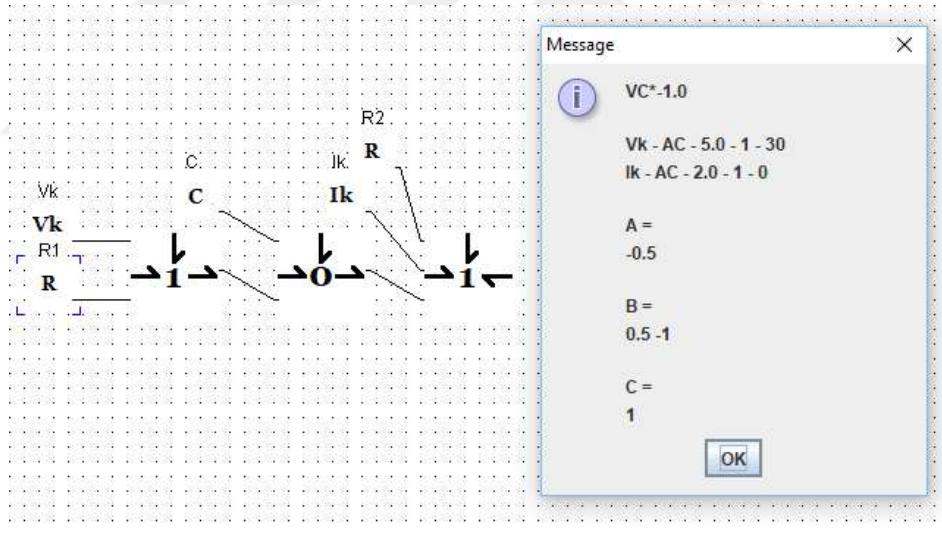
5.5. Örnek 5

Bu örnekte kullanılan devre Şekil 5.23.'de gösterilmiştir. Devrede 2 tane AC kaynak bulunmaktadır. Devrede eleman değerleri $R_1=2\Omega$, $R_2=5\Omega$, $C=1F$; kaynak değerleri $V_k = 5 \cdot \sin(t + 30)$, $I_k = 2 \cdot \sin(t)$; başlangıç değeri $V_C(0) = 1V$ olarak belirlenmiştir.



Şekil 5. 24. Örnek 5'e ilişkin devre şeması

Devrenin BGM ve program çıktıları Şekil 5. 25.'de görünmektedir.

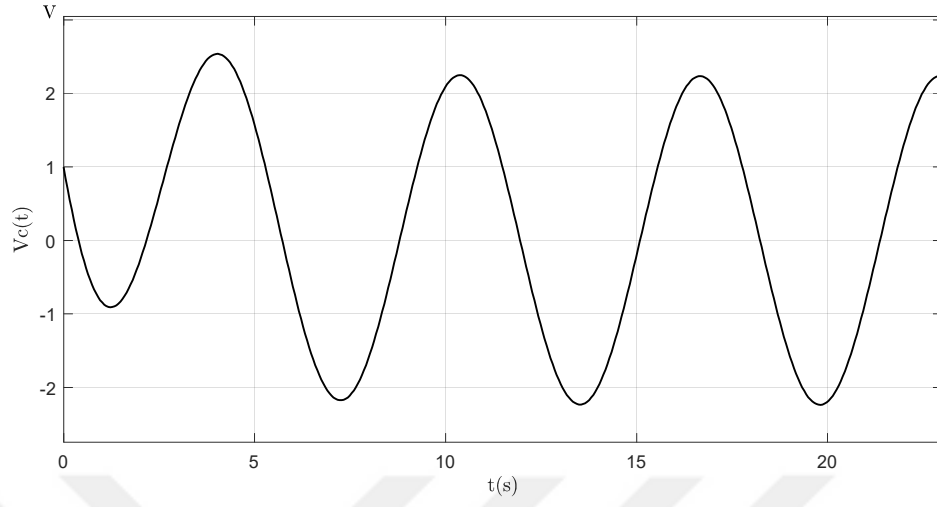


Şekil 5. 25. Örnek 5'e ilişkin BONAN program çıktıları

Program tarafından, (5.9)'da verilen durum denklemi elde edilir.

$$\dot{V}_C = -0.5 \cdot V_C + \begin{bmatrix} 0.5 & 1 \end{bmatrix} \cdot \begin{bmatrix} V_k \\ I_k \end{bmatrix} \quad (5.9)$$

Örnek 5'e ilişkin durum değişkeninin değişimi Şekil 5. 26'da verilmiştir.



Şekil 5. 26. Örnek 5'e ilişkin kondansatör geriliminin değişimi

6. SONUÇLAR

Mühendislik alanında dinamik sistemlerin modellenmesi ve benzetimi için grafiksel bir yaklaşım metodu olan BG yönteminin, herhangi bir karmaşık yapıli sistem modelinin geometrik ve gözleme dayanan bir yolla doğrudan sisteme bakarak elde edilmesi klasik graf tekniklerine göre üstünlüğünü ortaya koymaktadır. Ayrıca BG yönteminde, farklı enerji ortamlarındaki sistemler birleşik bir ortamda tasvir ve analiz edilebilirler. Bu avantajları göz önüne alındığında; BG yönteminin yazılımsal olarak desteklenmesi ve sistem tasarımcılarına kolaylıklar sağlaması amacıyla bu tez çalışmasında ortaya konan bilgisayar programı, elektriksel sistemlerin analizi için geliştirmeye açık bir örnek niteliğindedir.

Tezde oluşturulan bilgisayar programı, Netbeans geliştirme ortamında Java dili ile yazılmıştır. Program, temelinde lojik işlemler yapan açık kaynak kodlu LogicSim programının, devrelerin BGM'ni kullanarak analizini yapan bir yapıya dönüşmesiyle ortaya çıkmıştır. Amacına uygun olarak menüler ve eleman çıktıları tamamıyla değiştirilmiş, arayüz yapısının temeli büyük oranda eski formatında kalmıştır. Yazılımların BGM'ni giriş alacak şekilde değiştirilmesi zor bir süreç olmakla birlikte, Java dilinin sağladığı kolaylıklar ve nesne tabanlı bir yapıda oluşu, bu tez çalışmasında oldukça fayda sağlayan bir unsur olmuştur.

Programdan elde edilen veriler .m uzantılı Matlab dosyasına aktararak çözüm aşamasında kullanılmıştır. Buna ilaveten, grafik çıktılarının elde edilmesi de sağlanmıştır. Matlab ortamının özellikle çözüm kısmındaki üstünlükleri, oluşturulan programda yazılım sürecinin ksalmasını sağlamıştır.

Bölüm 5'de verilen örneklerle; BONAN programında BGM'nin oluşum aşamaları, devreye ilişkin tüm verilerin girişleri, durum ve çıkış denklemleri, Matlab ortamında gerçekleştirilen çözüm sonuçları gösterilmiştir. Sonuçların doğru olduğu gözlenmiştir. Bu tezde oluşturulan BONAN programı ile BGM'ni kullanan bilgisayar programlarına alternatif olabilecek bir yazılımın temeli oluşturulmuştur.

Gelecek çalışmalarda, elektrik devreleri ile farklı enerji ortamlarına ilişkin elemanlar programda tanımlanabilir. Ayrıca, çözüm aşamasında Matlab ortamına duyulan gereksinim ortadan kaldırılabilir.

KAYNAKLAR

- [1] **L.O. Chua, C.A. Desoer and E. S. Kuh**, Lineer and NonLineer Circuits, Mc Graw-Hill., New York, 1987.
- [2] **H. M. Paynter**, Analysis and Design of Engineering Systems, M.I.T. Press, Cambridge, Mass, 1961.
- [3] **M. Poyraz, Y. Demir, M. Koksall**, Lineer Olmayan Sistemlerin Bond Graf Modelinden Yararlanarak Durum ve Çıkış Denklemlerini Veren bir Algoritma, 4. Bilgisayar ve Haberleşme Sempozyumu, Elazığ, 1999.
- [4] **A.C. Umarikar, T. Mishra and L. Umanand**, Bond Graph Simulation and Symbolic Extraction Toolbox in MATLAB/Simulink, Centre for Electronic Design and Technology, Indian Institute of Science, Bangalore 560 012, India, 2005.
- [5] **A. Gulten**, Bağlaç Diyagramı Tekniğini Kullanarak Anahtarlı Devrelerin Modellenmesi ve Simulasyonu için bir Program, Doktora Tezi, Elazığ, 2003.
- [6] **D. Karnopp and R. Rosenberg**, System Dynamics: A Unified Approach, Wiley-Interscience, 402 pp., 1975.
- [7] **G.S. Ohm**, Ann. Phys., 6 (1826) 459.
- [8] **G.S. Ohm**, Ann. Phys., 7 (1827) 45,117.
- [9] **G.S. Ohm**, Die Galvanische Kette Mathematische Bearbeitet, Berlin, 1827.
- [10] **G.R. Kirchhoff**, Ann. Phys. Chem., 64 (1845) 497.
- [11] **G.R. Kirchhoff**, Ann. Phys. Chem., 72 (1847) 497.
- [12] **W.K. Clifford**, Am. J. Math., 1 (1878) 126.
- [13] **W.K. Clifford**, ... Theory of Graphs, Macmillan, London, 1881.
- [14] **A.B. Kempe**, Proc. London Math. Soc., 17 (1887) 107.
- [15] **A.B. Kempe**, Phil. Trans., 177, (1886) 1.
- [16] **C.S. Peirce**, The Monist, 7 (1897) 161 – Collected Papers, 3.456ff, Harvard, 1931-.
- [17] <http://www.me.utexas.edu/~longoria/paynter/hmp/Bondgraphs.html>
- [18] **H. M. Paynter**, , An Epistemic Prehistory of Bond Graphs, Elsevier Science Publishers B.V., North Holland, 1992.

- [19] **D.C. Karnopp, R.C. Rosenberg**, Analysis and Simulation of Multiport Systems - The Bond Graph Approach to Physical Systems Dynamics, M.I.T. Press, 1968.
- [20] **D.C. Karnopp**, Bond Graph Models for Electromagnetic Actuators, J. of the Franklin Institute, 319(1/2), pp.173-181, 1985.
- [21] **D.C. Karnopp, D.L. Margolis**, Analysis and Simulation of Planar Mechanism Systems Using Bond Graph, J. of Mechanical Design, 101(2), pp.187-191, 1979.
- [22] **R.C. Rosenberg**, A User's Guide to ENPORT-4, John Wiley, New York, 1974.
- [23] **D.C. Karnopp, R.C. Rosenberg**, Introduction to Physical System Dynamics, McGraw-Hill, New York, 1983.
- [24] **A. J. Blundell**, Bond Graphs for Modelling Engineering Systems, Ellis Horwood Publishers, Chichester, United Kingdom, and Halsted Press, New York, 151p, 1982.
- [25] **P.C. Breedveld**, A Definition of the Multibond Graph Language, Complex and Distributed Systems: Analysis, Simulation and Control (S.G.Tzafestas and P.Borne, eds.), IMACS Trans. Scientific Computing, North-Holland, Vol.4, pp.69-72, 1986.
- [26] **V. Abadie and G. D. Tanguy**, Opened Loop Control of Switching Linear Systems, J. of the Franklin Institute, 330(5), pp.799-813, 1993.
- [27] **F.E. Cellier**, New Frontiers in Bond Graph Modeling, International Conference on Bond Graph Modeling ICBGM'93 (J.J.Granda and F.E.Cellier, eds.), San Diego, SCS Publishing, Simulation Series, Vol.25, No.2, pp.7-8, 1993.
- [28] **F.E. Cellier**, World Wide Web - The Global Library: A Compendium of Knowledge About Bond Graph Research, 1997 International Conference on Bond Graph Modeling and Simulation (ICBGM'97) (J.J.Granda and G.Dauphin-Tanguy eds.), Phoenix, SCS Publishing, Simulation Series, Vol.29, No.1, pp.187-191, 1997.
- [29] **J.J. Granda**, Bond Graph Modeling Solutions of Algebraic Loops and Differential Causality in Mechanical and Electrical Systems, Proceedings IASTED Applied Simulation and Modeling Conference, San Francisco, pp.188-193, 1984.

- [30] **J.U. Thoma**, Physical Structure For Efficient Simulation - An Interdisciplinary Bondgraph Perspective, Proceedings CISS'94, First Joint Conference of International Simulation Societies, Zürich, pp.21-28, 1994.
- [31] **J.U. Thoma**, Block Bond Graphs for Simulation of Nonlinear Systems: A Technical Note, Trans. of the Society for Computer Simulation, 2(3), pp.249-255, 1985.
- [32] **N. Sen**, Bond Graph Modeling of Dynamical Systems By Hamiltonian Analysis, Proceedings 5th Princeton Conference on Information Science and Systems, pp.542-543, 1974.
- [33] **R. Tapia and A. Medina**, Doubly-fed wind turbine generator control: A bond graph approach, Simulation Modeling Practise and Theory 53, pp.149-166, 2015.
- [34] **L. Cauffriez, S. Grondel, P. Loslever and C. Aubrun**, Bond Graph modeling for fault detection and isolation of a train door mechatronic system, Control Engineering Practice 49, pp.212-224, 2016.
- [35] **I. Abdallah, A.L. Gehin and B.O. Bouamama**, Bond Graph for Online Robust Diagnosis Application: hydraulic system, IFAC PapersOnLine 50-1, pp.13538-13543, 2017.
- [36] **J.A.T. Machado and M.E. Mata**, A fractional perspective to the bond graph modelling of world economies, Nonlinear Dyn 80, pp.1839-1852, 2015.
- [37] **A. Vaz, K. Singh and G.D. Tanguy**, Bond graph model of extensor mechanism of finger based on hook-string mechanism, Mechanism and Machine Theory 91, pp.187-208, 2015.
- [38] **R.C. Rosenberg**, A User's Manual for ENPORT 1800, Department of Mechanical Engineering, Michigan State University East Lansing, Michigan, 1969.
- [39] <http://www.egr.msu.edu/~rosenberg/ENPORT/>
- [40] **J. J. Granda**, Computer Aided Modeling Program (CAMP): A Bond Graph Preprocessor for Computer Aided Design and Simulation of Physical Systems Using Digital Simulation Languages, Ph.D. Dissertation, Dept. of Mechanical Engineering, University of California Davis, 1982.
- [41] www.bondgraph.com

- [42] **A. Mukherjee and A.K. Samantaray**, System Modeling through Bond graph Objects on SYMBOLS 2000, Confereance Paper , India, 2001.
- [43] <http://www.symbols2000.com>
- [44] **J.F. Broenink**, 20-sim software for hierarchical bond graph/ block-diagram models, Simulation Practice and Theory 7, pp.481-492, 1999.
- [45] **M. Delgado and C. Pichardo**, Use of MATLAB and 20-sim to simulate a flash separator, Simulation Practice and Theory 7, pp.515-530, 1999.
- [46] <http://www.20sim.com>
- [47] **J.F. Broenink**, Object-oriented modeling with bond graphs and Modelica, ICBGM'99 International Conference on Bond Graph Modeling and Simulation San Francisco, CA, 1999.
- [48] **F.E. Cellier and R.T. McBride**, Object-Oriented Modeling of Complex Physical Systems using the Dymola Bond-Graph Library,
- [49] **Tevfik Kızılören**, Java ve Java Teknolojileri, İstanbul, Haziran-2010.
- [50] http://www.tetzl.de/java_logic_simulator.html
- [51] **Yakup Demir**, Lineer Olmayan Devrelerin Durum Denklemlerinin Değişik Bölgelerde Bulunması ve Bölgeler Arası Geçişler için Anahtarlama Denklemlerinin Hesaplanması, Doktora Tezi, Elazığ, 1993.

EK -1
KODLAMADA KULLANILAN
DEĞİŞKENLER

ne	: Devrenin toplam eleman sayısı
nd	: Devrenin toplam düğüm sayısı
dugummatris[][]	: $nd \times ne$ boyutlu elemana ilişkin düğümler matrisi
AnaMatris[][]	: $nd-1 \times ne$ boyutlu referansı kaldırılmış düğümler matrisi
DalMatrisi [][]	: Dal seçilen elemanlara ilişkin oluşturulan matris
KirisMatrisi [][]	: Kiriş seçilen elemanlara ilişkin oluşturulan matris
DalElemani []	: Dal seçilen elemanların indislerinin sıralandığı dizi
KirisElemani []	: Kiriş seçilen elemanların indislerinin sıralandığı dizi
x []	:Tüm elemanların dal ve kiriş olarak ayrıldıktan sonra oluşturduğu sıra dizisi
TemelKesitleme[][]	:Dal matrisi ve Kiriş matrisinin cebirsel işlemlerinden sonra oluşturulan eleman akımlarının katsayısını oluşturan matris
TemelCevre [][]	:Dal matrisi ve Kiriş matrisinin cebirsel işlemlerinden sonra oluşturulan eleman gerilimlerinin katsayısını oluşturan matris
EKD []	: Elemanlara ilişkin karakter dizisi
EDD []	: Elemanlara ilişkin değerler dizisi
AKD []	: Elemanlara ilişkin akımlar karakter dizisi
GKD []	: Elemanlara ilişkin gerilimler karakter dizisi
PED []	: Pasif elemanlar karakter dizisi
AED []	: Aktif elemanlar karakter dizisi
PEOKS []	: Pasif elemanların önündeki kaynak sayısının tutulduğu dizi
naktif	: Aktif eleman sayısı
npasif	: Pasif eleman sayısı
DDI []	: Durum değişkenlerinin $x[]$ sıra dizisinde bulunma indislerini tutan dizi
DDE []	: Durum değişkeni elemanlarını barındıran dizi
DDK []	: Durum değişkeni elemanının karakterler dizisi
DDKT []	: Durum değişkeni karakterlerinin türevlerini barındıran dizi
AGDD []	:Elemanlara ilişkin akımların, gerilimlerin ve durum değişkenlerinin türevlerinin bulunduğu dizi
DDBM [][]	: $(npasif+ne) \times ((2 \times ne)+nd)$ boyutlu durum değişkenlerini bulmada kullandığımız matris

DDSI [] []	:Durum deęiřkenlerinin AGDD[] dizisinde bulunma indislerini tutan dizi
KI []	:Kaynakların AGDD[] dizisinde bulunma indislerini tutan dizi
DDM [] []	:Durum deęiřkenlerinin deęerlerini DDBM [] [] matrisinden alan (npasif+ne)*ndd boyutlu matris
KM []	:Kaynakların deęerlerini DDBM[] [] matrisinden alan (npasif+ne)*naktif boyutlu matris
SSI []	:DDBM [] [] matrisinden silinecek durum deęiřkenleri ve kaynak sřtunlarının indis deęerlerini barındıran dizi
DDKM [] []	:Durum deęiřkenlerini bulmada kullanılacak, DDBM [] [] matrisinden silinen sřtunlardan sonra oluřan (npasif+ne)*(npasif+ne) boyutlu kare matris
A [] []	:Durum denklemlerini oluřturan durum deęiřkenlerine iliřin katsayı matrisi
B [] []	:Durum denklemlerini oluřturan kaynak deęiřkenlerine iliřin katsayı matrisi
C [] []	:Çıkıř denklemlerini oluřturan durum deęiřkenlerine iliřin katsayı matrisi

EK -2

ÇÖZÜM İÇİN KULLANILAN MATLAB KODU

```

function Xtam=BGtamcozum(A,B,x0,uT,uG)
syms s t u
[n m]=size(A);
Fis=s*eye(n,n)-A;
Fis=inv(Fis);
Xozt=ilaplace(Fis)*x0;
for j=1:size(uT)
    if uT(j)=='D'
        u(j)=uG(j,1);
    else
        u(j)=uG(j,1);
        u(j)=uG(j,1)*sin(uG(j,2)*t+uG(j,3));
    end
end
ozd=eig(A);
ts=max(real(ozd));
ts=6*abs(ts);
Us=laplace(u);
Xzors=Fis*B*Us;
Xzort=ilaplace(Xzors);
Xozt=simplify(Xozt);
Xzort=simplify(Xzort);
Xtam=Xozt+Xzort;
Xtam=simplify(Xtam);
figure(1)
ezplot(Xtam(1),[0 ts])
figure(2)
ezplot(Xtam(2),[0 ts])

```

ÖZGEÇMİŞ

Adı Soyadı: Fatma MURAT

Doğum Tarihi: 14 Mayıs 1994

Doğum Yeri: Keban/ Elazığ

Öğrenim Durumu:

Derece	Bölüm/ Üniversite	Yıl
Lisans	Elektrik-Elektronik Mühendisliği Bölümü, Fırat Üniversitesi	2012-2016
Yüksek Lisans	Elektrik Elektronik Mühendisliği Bölümü, Devreler ve Sistemler Anabilim Dalı, Fırat Üniversitesi	2016-2019

Yüksek Lisans Tez Başlığı ve Tez Danışmanı (Bitirme aşamasında):

Bond Graf Tekniğini Kullanarak Devrelerin Modellenmesi Ve Simülasyonu İçin Bir Bilgisayar Yazılımının Geliştirilmesi

Prof. Dr. Yakup DEMİR

Akademik Çalışma Alanları:

- Bond Graf Tekniği
- Devre Analizi
- Nesne Yönelimli Programlama