



Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

Bulanık Mantık ve Arduino Kullanarak Step Motorun Hız Kontrolü

Yusuf ALTUN*, Zuhal ÖZTÜRK, Hasret ÖZÜBERK

Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Düzce Üniversitesi, Düzce, TÜRKİYE

* Sorumlu yazarın e-posta adresi: yusufaltun@duzce.edu.tr

ÖZET

Bu çalışmada endüstriyel ve elektronik uygulamalarda ve robot teknolojilerinde kullanılan step motorun hız kontrolü gerçekleştirilmiştir. Tasarımda kullanılan Arduino IDE kod editörü ve derleyici olarak görev yaparak, derlenen programı karta yükleme işlemi sağlamıştır. Encoder sayısal bir elektrik sinyali üreten elektromekanik bir cihaz olması özelliğinden dolayı motorun kaç tur attığını algılamak için çalışmada yer verilmiştir. Hesaplama sonuçlarının listelendiği ekran olarak 2x16 display(LCD) ekran kullanılmıştır. Devreye eklenen butonlar ile hız artırma ve azaltma sağlanmıştır. Doğru malzeme seçimi modellenmesi ardından c programlama dili ve gerekli platformlar kullanılmıştır. Ares devre şeması proteus isis programı ile oluşturulmuştur. Gerekli işlemler yapılarak bakır plaket üzerine yapıştırılmıştır. Sanal ortamda denenen devre kablo ve test aşamalarının tamamlanmasıyla gerçek ortama geçirilmiştir. Bu sayfa bulanık mantık kullanılarak, girişe uygulanan işaret ile birlikte istenilen çıkışı veren step motor hız kontrolünü anlatmaktadır.

Anahtar Kelimeler: *Arduino, Step Motor, Bulanık mantık, Encoder*

Step Motor Control Using Fuzzy Logic and Arduino

ABSTRACT

In this study, step motors speed control achieved which is used industrial and electronic applications and robot technology. Arduino IDEAL, which is used as code editor and compiler, has provided loading process of program compiled. Encoder, what generates digital electrical signals, is electromechanical machine. In this study, ENCODER is used for understanding how many rolls have been done by motor. 2x16 display (LCD) screen has been used as the screen which calculation results to be listed. Increase-decrease rate is provided by the buttons which were added to circuit. After modelling choosing the right materials, c programming language necessary platforms has been used. Ares circuit plan is formed with Proteus Isis program. After necessary Processing, Ares circuit plan has been pasted on a copper plate. This study has been examined in virtual platform thank has been put into practice after cable and test stages. This page tells the speed control of stepper motors which give desired output by signal applied to input, by using fuzzy logic.

Keywords: *Arduino, Stepper Motor, Fuzzy logic, Encoder*

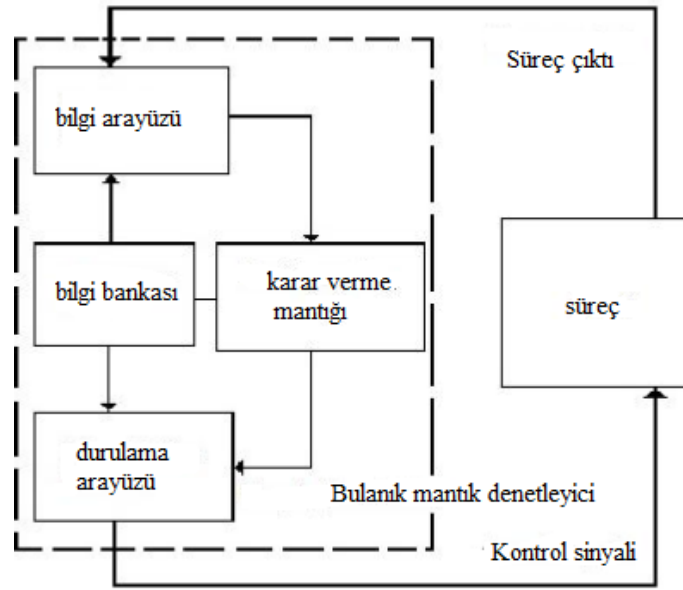
I. GİRİŞ

BULANIK mantık, 1961 yılında Lütü Aliasker Zade'nin yayınladığı bir makalenin sonucu oluşmuş bir mantık yapısıdır. 1965 tarihli orijinal makalesinde yayınlanmasından sonra belirsizlik içeren sistemlerin incelenmesi yeni bir boyut kazanmıştır. 1965 de ortaya atılmasına rağmen, bulanık küme kavramı ancak 1970'li yılların ikinci yarısından sonra kullanılmaya başlanmıştır [1]. Motor hız kontrolü her zaman birçok araştırmacı ve bilim adamının ilgi alanı olmuştur. İhtiyaç duyulan bu motor hız kontrolü sayısız uygulamalarda hayatımız da yer almaktadır. Bu yüzden, motor kontrolünü içeren birçok alanda ihtiyaç duyulmuştur [2-9] Mevcut bulanık mantık geliştirme düzenekleri kullanılıp, bulanık kümeler ve bulanık 'If-Then' kuralları bu uygulamalarda kullanılıp geliştirilerek hız hataları en küçük değere indirilmiştir. Bulanık küme, küme'ye aitlik derecesi üyelik değeri ile tanımlanmış olan kümeyi ifade eder. Uygun üyelik fonksiyonunun belirlenmesi, kontrol performansını oldukça etkilemektedir. Adım motorları (Step Motorlar), girişlerine uygulanan darbe dizilerine karşılık analog dönme hareketi yapan fırçasız, genellikle kalıcı mıknatıs kutuplu DC motorlardır. Bu motor tipinin ilk çalışma ilkeleri 1930'lu yıllarda açıklanmış, daha sonraki yıllarda gemilere ve çeşitli savaş ağırlıklı teçhizata uyarlanmıştır. 1944-1957 yılları arasındaki dönemde kapalı çevrim servo sistemleri ile yaygın bir kullanım alanı bulan adım motorları 1960'lı yıllarda transistörlerin kullanımı ile daha da artmış ve 1970'li yıllardan itibaren dijital elektroniğin gelişmesine paralel olarak oldukça yaygınlaştırmıştır [9]. Adım açısı motorun yapısına bağlı olarak 180°, 90°, 45°, 18°, 7.5°, 1.8° veya daha değişik açılarda olabilir. Adım motorunun hızı, uygulanan darbelerin frekansı değiştirilerek kontrol edilebilir [3].

Daha önce bulanık mantık kullanarak hız kontrolü yapılan çalışmalardan step motor olmadan DC motor ile örneği "DC şönt motorun bulanık mantık ile hız kontrolü" yapılmıştır [7]. Bu motorlarda hız, mil yüküne bağlıdır. Eğer yük fazlaysa rotor düşük hızda döner. Yük az olduğunda da motor daha hızlı dönecektir. Şönt motorların hız kontrolü alan sargısına seri bağlanan bir direnç sayesinde yapılır. DC şönt motor mil yüküne bağlı olarak hızlanma ve yavaşlama gösterdiğinden step motor tercih edilmiştir. Benzer bir başka hız kontrol çalışması olan "PLC Tabanlı AC Servo motorun Konum ve Hız Denetimi" dir. Programlanır lojik kontrolör PLC, endüstriyel otomasyon sistemlerinin kumanda ve kontrol devrelerini gerçekleştirmek için kurulan donanım birimlerini uygun şekilde I/O giriş-çıkış (input-ouput) birimleri sunan ve kontrol yapısına uygun iletişim arabirimi sunan iç yapısına yazılan programı kullanıcı isteğine göre otomatik olarak yerine getiren önemli bir kontrol ünitesidir [8]. PLC Endüstri şartlarına daha uygundur mikrodenetleyici sayılan arduinoya ya nazaran maliyet olarak çok daha pahalıdır biraz daha profesyonellik gerektirir bu sebepten çalışmada arduino tercih edilmiştir [10, 11]. Bir başka çalışma olarak "PID ve Bulanık Mantık ile DC Motorun Gerçek Zamanda STM32F407 Tabanlı Hız Kontrolü" dür [9]. PID türevsel ve integral oranlarıyla katsayı bilgilerine ulaşılması ve uygun değerlerde sabit tutulması işlemi olduğundan hız kontrolünü istediğimiz değerde tutabilmek için hesap yapmak gerekir ve katsayıları sürekli uygun değerlerde tutmak gerekir. PID sistemi endüstriyel sistemlerde daha çok kullanılır hesaplamalarla uğraşmak zaman almakla beraber yorucu hale gelir bu yüzden kod kullanarak işlem yapmak daha kolay olur. Arduino ile yapıldığında sadece kodlarla ufak değişiklikler yaparak hız değerlerini değiştirmek mümkündür. Bu sebeplerden çalışmada arduino tercih edilmiştir. Yine bir başka hız kontrol çalışması "Low Speed Control of AC Servo Motors in No-Load Condition"dir [12]. Step motorlar servo motorlara göre daha kolay kontrol edilebilir. Servo motor daha çok profesyonel alanlarda tercih edilir. Çünkü step motora göre daha pahalıdır. Bu sebeplerden devrede step motor tercih edilmiştir. Yapılan bir başka bulanık mantık ile hız kontrol çalışması olarak "Induction Motor Speed Control using Fuzzy Logic Controller"dır [13].

II. YÖNTEM

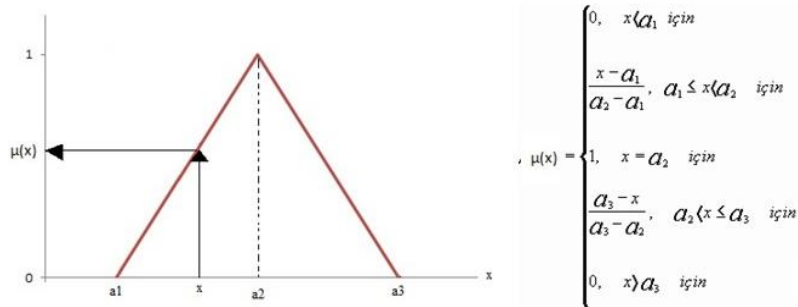
Devrede dışarıdan girilmiş olan hız değerine göre step motor girilen değere göre ayarlanan devirde gezmeye çalışır. Step motor devri 5 ise ve girilen hız değeri 2 ise step devri 2'ye indirmek için hız değerini bulanık mantık ile ayarlar. Bulanık mantık standart lojikte bulunan sıfır ve bir seviyeleri yerine çıkışların ve girişlerin birden çok üyelik fonksiyonlarına üye olabildiği bir kontrol yöntemidir. Bir bulanık mantık denetleyicisi dört temel bileşen olarak verilebilir; Bulanıklaştırma ara yüzü, bilgi-tabanı, karar verme mantığı, durulaştırma ara yüzü [14]. Şekil 1'de bu yapı verilmiştir. Sırasıyla bu metot ve adımlar uygulanarak tasarım gerçekleştirilmiştir.



Şekil 1. Bulanık mantık denetleyici yapısı

A. BULANDIRMA

Bulanık mantık kontrol yapabilmesi için öncelikli olarak girişlerin bulandırılması gerekmektedir. Bulandırma işlemi üyelik fonksiyonları kullanarak yapılmaktadır. $N_{referans}$ ve $n_{encoder}$ farkı alınarak e (hata) bulunur, bulunan hata değerinden bir önceki hata değeri çıkarılarak de (hata değişimi) bulunur. e_{hata} ve $de_{hata_değişimi}$ K_1 normalizasyon katsayısıyla çarpılarak 'Bulandırma' ara birimine gönderilir. Bulandırma ara biriminde μ_e , μ_{de} değerleri Şekil 2'deki formüle göre hesaplanarak bulanık kontrol kurallarıyla (1) eşitliğinden faydalanarak μ_{du} bulunur.



Şekil 2. Üyelik değeri formülü

$$\mu_{du} = \mu_e \cap \mu_{de} = \min(\mu_e, \mu_{de}) \quad (1)$$

B. DURULAŞTIRMA

‘Durulaştırma’ ara biriminde (2) eşitliğinden yararlanılarak bulanık kontrol çıkışı olan DU elde edilir. DU değeri K2 normalizasyon katsayısı ile çarpılarak yeni DU değeri hesaplanır ve giriş devir değerinin bir önceki değeri ile toplanarak yeni u giriş deviri elde edilir. Bu şekilde referans değerine ulaşılan kadar kapalı döngü sistemi devam etmektedir. Bu çalışmadaki sistem için belirlenen normalizasyon katsayıları K1=1, K2=1’dir. Elde edilen DU değeri ile u giriş devrinin bir önceki değeri ile toplanarak bulanık kontrolün çıkışı olan u elde edilir. Döngü referans değerine ulaşılan kadar bu şekilde devam eder.

$$DU = \frac{du1 \cdot \mu_{du1} + du2 \cdot \mu_{du2} + du3 \cdot \mu_{du3} + du4 \cdot \mu_{du4}}{\mu_{du1} + \mu_{du2} + \mu_{du3} + \mu_{du4}} \quad (2)$$

C. ÜÇGEN ÜYELİK FONKSİYONU

Üyelik fonksiyonu olarak Şekil 3’te olduğu gibi üçgen fonksiyonu kullanılmıştır. Üçgen üyelik fonksiyonunun kullanılmasının temel sebebi ise işlemci tarafında hesap yükünün diğer fonksiyonlara göre daha az olmasıdır. Bu özelliği sayesinde üçgen fonksiyonu gerçek zamanda çalışma için diğer fonksiyonlara göre daha uygun olmaktadır. Uygun üyelik fonksiyonunun belirlenmesi, kontrol performansımızı oldukça etkilemektedir.

Tablo1. Bulanık Küme

Step Motor hız değerleri	Durum
1-3	Yavaş hız değeri
2-5	Orta hız değeri
4-6	Hızlı



Şekil 3. Üyelik Fonksiyonu değerleri

D. HIZIN BULANIK DEĞİŞKEN İLİŞKİSİ

Bulandırma: Bulandırma ara biriminde kontrol edilecek sisteme göre tasarlanan üyelik fonksiyonları bulunmaktadır. Bu çalışmada üçgen tipi üyelik fonksiyonu kullanılmıştır. Sistemin giriş değerine karşı, çıkış değeri dikkate alınarak tasarlanmıştır.

Tablo2. Tasarlanan kontrol kuralı

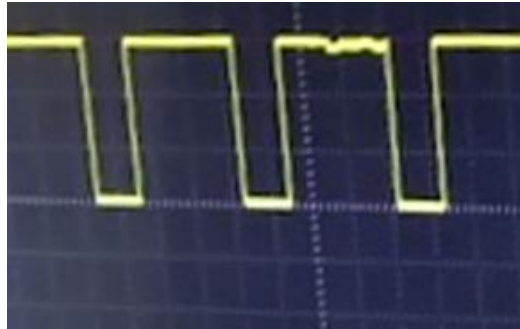
<i>e\de</i>	NB	NO	NK	S	PK	PO	PB
NB	NB	NB	NB	NB	NO	NK	S
NO	NB	NB	NB	NO	NK	S	PK
NK	NB	NB	NO	NK	S	PK	PO
S	NB	NO	NK	S	PK	PO	PB
PK	NO	NK	S	PK	PO	PB	PB
PO	NK	S	PK	PO	PB	PB	PB
PB	S	PK	PO	PB	PB	PB	PB

Burada tasarlanan kontrol kuralları yer almaktadır. e (hata) ve de (hata değişimi) sembollerinin üyelik fonksiyonlarındaki değerlerine göre belirlenen, μ_e ve μ_{de} kontrol kuralları elde edilmiştir. μ_e , μ_{de} nicel çıkış değerimizi elde etmek için kesişim ve birleşim fonksiyonları kullanılmıştır.(3) eşitliğinde ki kesişim fonksiyonuyla belirlenen μ_{de} değerleri durulaştırma ara biriminde kullanılır.

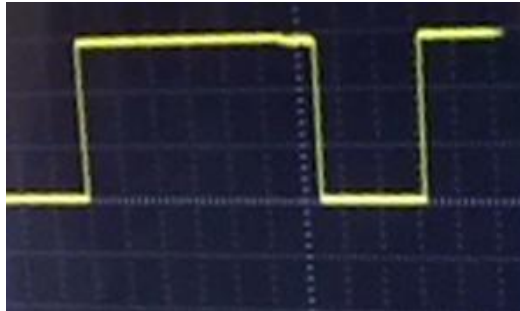
$$\mu_{de} = \mu_e \cap \mu_{de} = \min(\mu_e, \mu_{de}) \quad (3)$$

III. BULGULAR ve TARTIŞMA

A. UYGULAMA



Şekil 4. Hız artınca kare dalga sayısı artar



Şekil 5. Hız yavaşlayınca kare dalga sayısı azalır

Şekil 4'te görüldüğü üzere motor hızı artırıldığı zaman görüntüye sığan kare dalga sayısı artmıştır. Kare dalga sayısı arttıkça devir sayısı artar. Şekil 5'te hız yavaşlayınca tam tersi durum söz konusudur ekrana sığan kare dalga sayısı azalır ve kare dalga eni genişler.



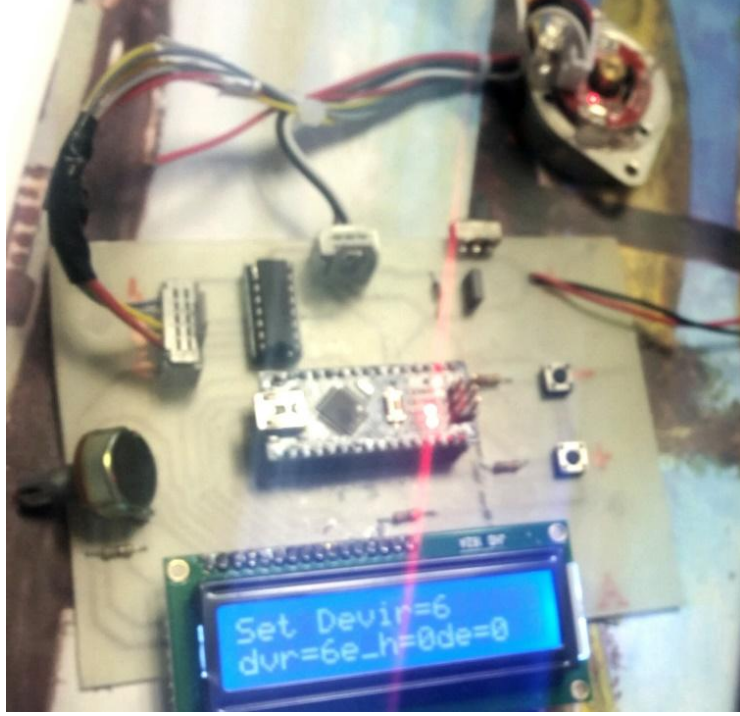
Şekil 6. Set devire giriş olarak 2 verilmesi

Şekil 6 değişik referans değerini göstermektedir. Motor hızı 5 devirken, 2 devrine düşürülmesi durumunda motor hızı azalarak 2 değerine ulaşacaktır. Şekil 7'de görüldüğü gibi girişe uygulanan işaret bulanık mantık ile istenilen çıkışı üretmiştir.



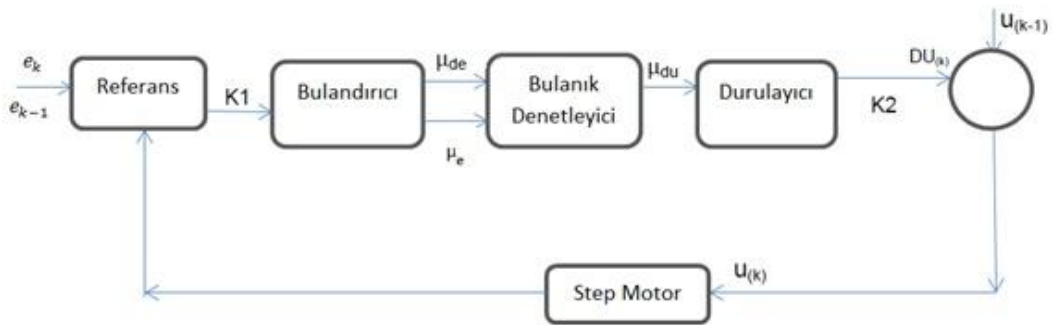
Şekil 7. Hız yavaşlayarak istenilen 2 değerine ulaşması

Kontrol etmek istenen hız değeri step motorun hız değeri ile karşılaştırılmaktadır. Doğru sonuç elde edilerek istenilen hız değerine gelene kadar kontrol etmektedir. Hız değerleri 1 ile 6 arasında değişmektedir. Bulanık mantık ile kontrol için aralık Tablo1’de olduğu gibi tanımlanmıştır. Kural çıkartımı yapılmış olup belirlenen kurallara göre sistemin kontrolü sağlanmıştır.



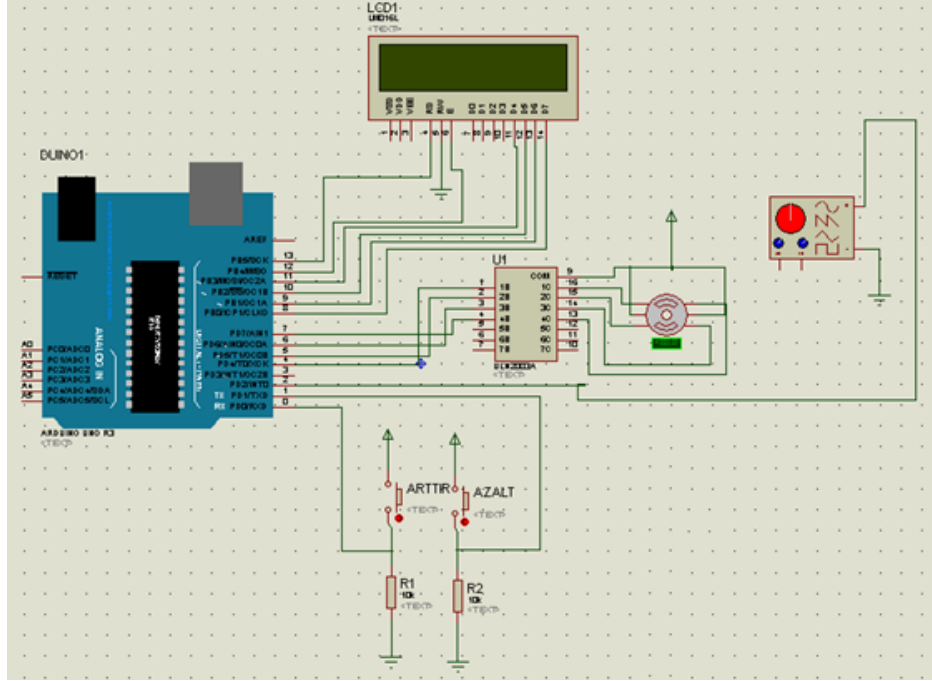
Şekil 8. Devre görüntüsü

Devrenin son hali olan Şekil 8’de görüldüğü üzere; LCD ekranda devrenin çalıştırılmasıyla ayarlanan hız devri, motorun o anki devri, ikisi arasındaki hata değeri ve değişim yazmaktadır. Bulanık mantık kurallarıyla hesaplanarak elde edilen bu değerler gösterilmektedir.



Şekil 9. Geliştirilen bulanık mantık sisteminin blok diyagramı

Şekil 9 kontrol devresinin özeti niteliğinde blok diyagramını göstermektedir. Girilen referans değeriyle başlayan, bulandırma işlemi üyelik fonksiyonları kullanarak yapılan, denetleyici ve durulayıcı gibi bir dizi işlem ve kurallardan geçilerek elde edilen sistem kontrolüdür.



Şekil 10. Proteus programında çizilen devre bağlantı yolları

Devre çizimi ve testi proteus programında gerçekleştirilmiştir. Devrenin bağlantıları Şekil 10'da verildiği gibi yapılmıştır. Bu bağlantıları eklerken elemanların çalışma mekanizmaları ve kullanım şekillerine hakim olmak gerekmektedir.

IV. SONUÇ

Bu çalışmada klasik mantık yerine bulanık mantık tercih edilmiştir. Klasik mantıkla yapılan bir devrede kullanıcı kontrolü gerekmektedir. Bulanık mantıkta ise kullanıcı kontrolüne gerek yoktur. PID gibi klasik kontrollerinde yaygın bilinen eksikleri de mevcuttur. Bulanık mantık sistemi kendi kendini kontrol edebilmektedir. Bu sayede bütün olasılıklara göre sistem çözüm üretmektedir. Üretilen bu çözümler sayesinde sistem hızına ve çalışmasına karar vermektedir. Sistemin bu kararları vermesinde bulanık kontrol sistemi kuralları etkili olmuştur. Sistemin hassasiyetini artırmak için bulanıklaştırma aşamasına daha çok olasılık eklenebilir. Sistemin hız değerleri arasındaki hız geçiş sürelerinin zamanları azaltılabilir. Sonuç olarak, bulanık mantıkla step motorunun devir kontrolü başarıyla gerçekleştirilmiştir.

V. KAYNAKLAR

- [1] L.A. Zadeh *IEEE Transactions on Systems, and Cybernetics* **3(1)** (1973).
- [2] T.A. Khan, S.N. Adnan, M.H. Hany, *Speed control of hybrid stepper motor by using an adaptive neuro fuzzy inference system*, **IEEE 2014 6th International Conference on Electronics, Computers and Artificial Intelligence**, (2014).

- [3] A.S. Gong, B. Bin He, *LabVIEW-base automatic rising and falling speed control of stepper motor*, **IEEE International Conference on Electrical Machines and Systems**, (2009).
- [4] J.T. Boys *IEE Journal on Electric Power Applications* **2(1)** (1979) 27.
- [5] M. Bendjedia, Y. Ait-Amirat, B. Walther, A. Berthon, **European Conference on Sensorless control of hybrid stepper motor**, **Power Electronics and Applications**, (2007) 1.
- [6] H. Shuanghui, L. Yong, H. Minghui, *Study on a novel absolute magnetic encoder*, **IEEE International Conference on Robotics and Biomimetics**, (2009) 1773.
- [7] M. Zile, *DC şönt motorun bulanık mantık ile hız kontrolü*, **III. Otomasyon Sempozyumu**, Denizli-Türkiye, (2005).
- [8] İ. Coşkun, M.F. Işık *Otomasyon Dergisi* **(3)** (2009) 232.
- [9] F. Köş, K. Kaplan, H.M. Ertunç, *PID ve bulanık mantık ile DC motorun gerçek zamanda STM32F407 tabanlı hız kontrolü*, **Otomatik Kontrol Ulusal Toplantısı**, (2013) 26.
- [10] G. Lastovicka-Medin, M. Petric, *Embedded lab: Arduino projects in science lessons*, **4th Mediterranean Conference on Embedded Computing**, (2015) 284.
- [11] C. Taşdemir, *Arduino*, Dikeyksen Yayıncılık, (2012).
- [12] I. Kardan, A. Akbarzadeh, K. Kamali, *Low speed control of AC servo motors in no-load condition*, **RSI/ISM International Conference on Robotics and Mechatronics**, (2014).
- [13] V. Chitra, R.S. Prabhakar *World Academy of Science, Engineering and Technology* **23** (2006) 17.
- [14] L.A. Zadeh *Information Sciences* **8** (1975) 199.