به نام خدا

کنترل هوشمند موتور *DC* با استفاده از *ANFIS*

گردآوری

احمدرضا سعیدی

کلمات کلیدی

فازی-عصبی، کنترل کننده *PID*، *Simulink* ، موتور *DC*، *ANFIS*،

چکیده

این پایان نامه به کنترل سرعت موتور***DC*** با استفاده از سیستم کنترل فازی –عصبی تطبیقی (***Adaptive Neuro- Fuzzy Inference System***) می پردازد. این کنترل کننده بر اساس کنترل فازی – عصبی تطبیقی پایه گذاری شده است و هدف کلی آن کاهش خطای حالت ماندگار (***stady state error***) و زمان خیز (***rise time***) می باشد.

فهرست

چکیده

مزیت اصلی استفاده از موتور ***DC*** در جهان امروز قابلیت کنترل راحت سرعت و زاویه موتور آن است. خیلی از سیستم های کنترلی طراحی شده اند تا با کاهش اُورشورت ها و زمان نشست عملکرد بهتری را برای موتور ایجاد کنند. این پایان نامه به کنترل سرعت موتور***DC*** با استفاده از سیستم کنترل فازی – عصبی تطبیقی (***Adaptive Neuro- Fuzzy Inference System***) می پردازد. این کنترل کننده بر اساس کنترل فازی – عصبی تطبیقی پایه گذاری شده است و هدف کلی آن کاهش خطای حالت ماندگار (***stady state error***) و زمان خیز (***rise time***) می باشد.

یک کنترل کنندهٔ ***PID*** در این پایان نامه طراحی می شود تا براساس اطلاعات یادگرفته شده از سیستم توسط ***ANFIS*** سرعت موتور را کنترل کند. ما این سیستم کنترل را با نرم افزار ***MATLAB*** و با استفاده از قسمت ***Simulink*** آن شبیه سازی می کنیم.

عملکرد دو نوع سیستم کنترل ***PID*** و***ANFIS*** در گشتاورهای بار مختلف در این پایان نامه ارزیابی می شود.

اجرای عملی این سیستم در آزمایشگاه موتور ***DC*** با موفقیت بوسیلهٔ رابط ***Dspace*** انجام شده است.

در این پایان نامه نتایج حاصل از شبیه سازی کامپیوتری با نتایج عملی مقایسه می شوند.

فصل اول

۱. مقدمه

موتور ***DC*** در کاربردهای فراوانی استفاده می شود .به دلیل قابلیت کنترل راحت و عملکرد سریع موتور ***DC*** این موتورها در محدودهٔ وسیعی از سرعت ها قابل تنظیم می باشند.

کنترل کننده های سرعت که طراحی می شوند تا سرعت موتور***DC*** را برای انجام انواع وظایف کنترل کنند، بر دو نوع پیوسته *(آنالوگ)* و گسسته *(دیجیتال)* می باشند.کنترل کننده ها می توانند در انواع متناسب ***P*** (***Proportional***) ،متناسب – انتگرال گیر (***Proportional- Integral***) ، متناسب – انتگرال گیر – مشتق گیر ***(Proportional-Integral-Derivative)*** ، شبکه های عصبی ، منطق فازی و یا ترکیبی از اینها باشد.

در قمست *[1 و2]* کنترل کنندهٔ موتور با استفاده از کنترل کنندهٔ ***PID*** طراحی می شوند. مشکل کنترل کننده های ***PID*** تنظیم مشکل آن و داشتن اُورشوت می باشد. همچنین کارهای زیادی درزمینه کنترل موتورهای ***DC*** بااستفاده از شبکه های عصبی و ترکیب عصبی – ***PID*** انجام شده است. یک شبکهٔ عصبی می تواند با استفاده از مزیت های یادگیری ماشین و الگوریتم های یادگیری ، یک سیستم را کنترل کند. اما استفاده از یک شبکهٔ عصبی کنترل کننده برای کنترل یک سیستم زمان زیادی می گیرد.

این به خاطر آن است که در ابتدا نرم افزار کنترل کننده باید زمان زیادی را برای یادگیری از سیستم صرف کند.

کنترل کننده های منطق فازی نیز امروزه در بسیاری از کاربردهای کنترلی برای سیستم های پیچیده و غیرخطی توسعه یافته اند.

در قسمت *[5]* در مورد کنترل کننده های موتور ***DC*** با استفاده از منطق فازی بحث شده است. عیب استفاده این نوع سیستم کنترلی این است که این سیستم ها مبتنی بر قواعد « اگر – آنگاه » هستند که این قواعد را ما از توصیف سیستم بدست آورده ایم. چون ما در مورد رفتار اکثر سیستم اطلاعات زیادی نداریم نتیجه گرفتن از این نوع سیستم ها مشکل است.

کاری که در این پرورژه انجام شده استفاده از ترکیب شبکه های عصبی و منطق فازی در کنترل ماشین است. اما این کنترل کننده قبل از انجام کار کنترلی آن به دلیل داشتن شبکهٔ عصبی در آن در ابتدا نیاز

به زمانی برای یادگیری دارد . یک شبکهٔ عصبی دارای این مزیت است که می تواند رفتار سیستم را یاد بگیرد و حدس بزند .

منطق فازی، منطقی است بر اساس قواعد مبتنی بر دانش . قلب یک سیستم فازی یک پایگاه دانش بوده که از قواعد « اگر – آنگاه » فازی تشکیل شده است .این قواعد به صورت عبارت سادهٔ قابل فهم روزانه ساخته می شوند. این قواعد را افراد خبره که رفتار سیستم را کاملاً می شناسند با زبان طبیعی تعریف می کنند.

یک سیستم کنترلی فازی – عصبی تطبیقی از مزّیت های هر دو نوع سیستم کنترلی فازی و عصبی کمک می گیرد. این پایان نامه به بررسی کنترل سرعت موتور ***DC*** مبتنی بر ***ANFIS*** می پردازد. روش کنترلی آبشاری (***cascade control***) نیز برای کنترل موتور***DC*** در این پایان نامه بحث می شود. مدل ها در محیط ***Matlab/ simulink*** ساخته می شوند و مدل های ساخته شده با استفاده از رابط ***dSPACE5.0*** به سخت افزار وصل می شوند. سخت افزاری که برای این کار استفاده می شود شامل یک ***Lab- Kit Dc motor*** و یک بورد درایو الکتریکی می باشد. مشخصات موتور ***DC*** مورد استفاده قرار گرفته در این آزمایش به شرح زیر است.

ولتاژ ماکسیمم	۴۲*V*
سرعت	۴۰۰ ***rpm***
جریان نامی	*4/8* آمپر در هر فاز
توان نامی	*250* ولت
شرکت سازنده	***motorsoft***

یک منبع تغذیهٔ ***DC*** رگوله شده بوسیلهٔ بورد درایو الکتریکی ایجاد می شود که ولتاژ لازم برای درایو کردن موتور ***DC*** (معمولاً *42* ولت) را فراهم کند .برای اطلاعات بیشتر در مورد درایو بورد الکتریکی به پیوست ***B.2*** مراجعه کنید. در قسمت *1-1* این فصل مروری بر ***simulink*** و ***dSPACE*** خواهیم داشت. مدل ریاضی موتور ***DC*** و طراحی آن در***simulink*** در قسمت *2-1* بررسی می شود. روش کنترل آبشار (***cascade control***) درفصل *3-1* بحث می شود.

۱-۱. مقدمه ای بر simulink و dSPACE

مدل سازی ، طراحی و آنالیز کنترلر در ***Matlab/ simulink*** انجام می شود. شبیه سازی مدل ریاضی سیستم با استفاده از ***simulink*** انجام می شود.

Simulink کمک می کند تا مسائل را به صورت گرافیکی تحلیل کنیم که این کار با اتصال بلوکهای مختلف به هم و شکل دهی معادلات دیفرانسیل به صورت گرافها و نمودارها انجام می شود. ***Simulink*** شبیه نمونه برداری یک کامپیوتر دیجیتال از یک کامپیوتر آنالوگ عمل می کند. همۀبلوکها سیمولینک در کتابخانۀ آن موجود می باشند. وقتی مدل در سیمولینک ساخته شد و به کد ماشین تبدیل شد آمادۀ اجرا در پردازشگر ***DSP*** می گردد. ***DSPACE*** این امکان را فراهم می کند که کنترلر طراحی شده در سیمولینک را به راحتی در زمان واقعی در حال کارکرد واقعی آزمایش کرد. این کار با دانلود کردن مدل در ***DSP*** انجام می شود.

مزیت استفاده از ***DSP*** این است که می تواند مستقیماً با سیمولینک ارتباط برقرار کند .ویژگی اصلی ***DSPACE*** این است که دارای معماری باز ***(Open architect)*** می باشد که این امکان را فراهم می کند تا هر الگوریتم کنترلی روی هر ماشین استاندارد اجرا و آزمایش شود.

ویژگی دیگر***DSPACE*** قابلیت پیکربندی آن به صورت گرافیکی ***(graphical user interface)*** می باشد. مطالب بیشتر در مورد ***dSPACE*** در پیوست ***B.1*** موجود می باشد.

۱-۲. مدلسازی موتور DC

یک موتور ***DC*** آهنربای دائم نیروی الکتریکی را به نیروی مکانیکی تبدیل می کند. مدار الکتریکی یک موتور ***DC*** در شکل ۱.***1*** رسم شده است که در آن :

Va : ولتاژ تغذیه موتور ***DC***

Ra: مقاومت آرمیچر

B: ضریب دمپینگ

La: اندوکتانس

Ea : ولتاژ القایی

Wa: سرعت موتور

Kt : ثابت گشتاور

می باشد

Tw: گشتاور اصطکاک

Tw : گشتاور اینرسی

J: ممان اینرسی

Tl : گشتاور بار

Te : گشتاور الکترومغناطیسی

Ke: ثابت ولتاژ القایی

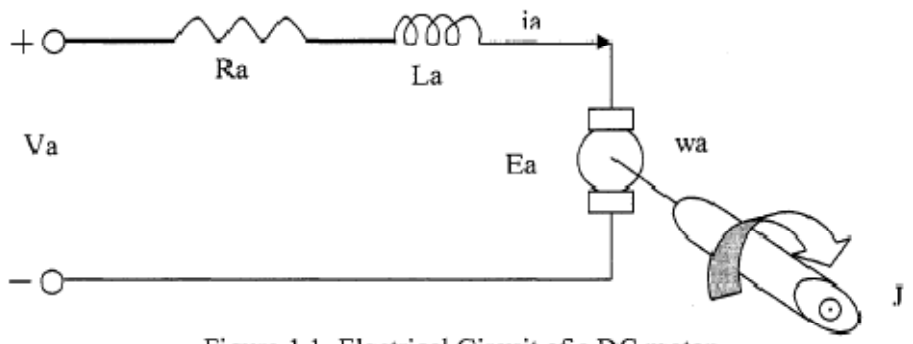

Figure 1.1. Electrical Circuit of a DC motor

۱-۳. مشخصات الکتریکی

با زدن یک ***KVL*** در مدارشکل ۱.*1* یک معادلهٔ دیفرانسیل از آن حاصل می شود.

$$V_a - V_{ra} - V_{la} - V_c = 0 \qquad (1.1)$$

با توجه به قانون اُهم داریم :

$$V_{ra} = i_a R_a \qquad (1.2)$$

ولتاژ دو سر سلف متناسب با مشتق جریان گذرنده از سلف نسبت به زمان می باشد که می تـوان نوشـت :

$$V_{la} = L_a \frac{di_a}{dt} \qquad (1.3)$$

ولتاژ القایی ***(back emf)*** را می توان به صورت زیر نوشت:

$$E_a = k_e \omega_a \qquad (1.4)$$

با جایگذاری روابط *(۱.۲)* و *(۱.۳)* و *(۱.۴)* در *(۱.۱)* نتیجه می شود :

$$V_a - i_a R_a - L_a \frac{di_a}{dt} - k_e \omega_a = 0 \qquad (1.5)$$

قانون بقای انرژی ایجاب می کند که جمع گشتاور های موجود در موتور باید برابر صـفر باشـد. بنـابراین داریم :

$$T_e - T_\omega - T_{\omega'} - T_l = 0 \qquad (1.6)$$

گشتاور الکترومغناطیس با جریان سیم پیچ آرمیچه متناسب است و می توان نوشت :

$$T_e = k_t i_a \qquad (1.7)$$

$$T_{ai} = J\frac{d\omega_a}{dt} \qquad (1.8)$$

$$T_\omega = B\omega_a \qquad (1.9)$$

با جایگذاری روابط (۱.۷) و (۱.۸) و (۱.۹) در (۱.۶) معادلهٔ دیفرانسیل زیر نتیجه میشود :

$$k_t i_a - J\frac{d\omega_a}{dt} - B\omega_a - T_l = 0 \qquad (1.10)$$

با مرتب کردن معادلات (۱.۵) و (۱.۱۰) به صورت معادلات حالت ، می توان نوشت :

$$\frac{di_a}{dt} = -\frac{R_a}{L_a}i_a - \frac{k_e}{L_a}\omega_a + \frac{V_a}{L_a} \qquad (1.11)$$

$$\frac{d\omega_a}{dt} = \frac{k_t}{J}i_a - \frac{B}{J}\omega_a - \frac{T_l}{J} \qquad (1.12)$$

معادلات (۱.۱۱) و (۱.۱۲) در فرم ماتریسی به صورت زیر در می آیند:

$$\frac{d}{dt}\begin{bmatrix} i_a \\ \omega_a \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{k_e}{L_a} \\ \frac{k_t}{J} & -\frac{B}{J} \end{bmatrix}\begin{bmatrix} i_a \\ \omega_a \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix}\begin{bmatrix} V_a \\ T_l \end{bmatrix}$$

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} i_a \\ \omega_a \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} V_a \\ T_l \end{bmatrix}$$

با اعمال تبدیل لاپلاس بر روی معادلات (۱.۱۱) و (۱.۱۲) خواهیم داشت :

$$I_a(s) = \frac{V_a(s) - E_a(s)}{R_a + sL_a} \qquad (1.13) \quad \text{where} \quad E_a(s) = k_e.\omega_a(s)$$

$$\omega_a(s) = \frac{T_e(s) - T_l(s)}{sJ + B} \qquad (1.14) \quad \text{where} \quad \begin{matrix} T_e(s) = k_t I_a(s) \\ k_t = k_e \end{matrix}$$

۱-۴. طراحی موتور DC در سیمولینک

از معادلات فوق بلوک دیاگرام موتور **DC** بدست می آید که در زیر نشان داده شده است.

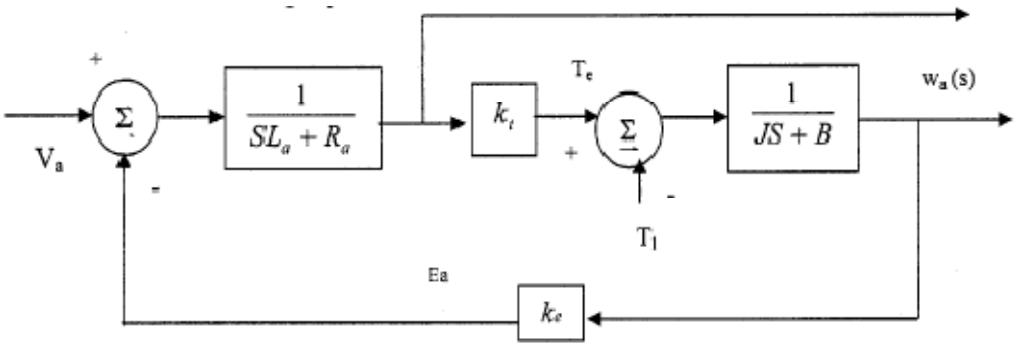

Figure 1.2. Block diagram representation of DC motor

بنابراین با ملاحظه بلوک دیاگرام فوق ما می توانیم بلوک دیاگرام موتور ***DC*** را در سیمولینک بسازیم که در شکل ۱.۳ نشان داده شده است:

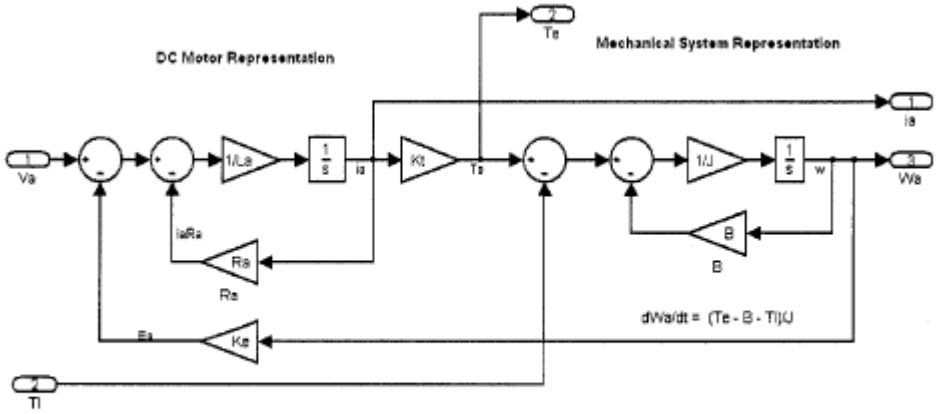

Figure 1.3. Simulink diagram of DC motor

پارامترهای موتور ***DC*** در پیوست ***A*** بحث شده است .

۱-۵. کنترل آبشاری موتور DC

یک سیستم کنترلی آبشاری شامل دو حلقه فیدبک است که برای کنترل سرعت موتور ***DC*** استفاده می شود و عملکرد سیستم را نسبت به سیستم کنترلی تک حلقه ارتقا می بخشد . حلقهٔ کنترلی داخلی برای کاهش پارازیت های روی حلقهٔ دوم استفاده می شود در حالی که حلقهٔ خارجی برای کاهش حساسیت متغیرهای مرحله اول نسبت به نوسانات بهره استفاده می شود.

سیستم کنترل آبشاری مورد بحث در این پایان نامه دو نوع کنترل کننده را با هم درگیر می کند .یکی کنترل کنندهٔ سرعت که برای کنترل جریان موتور ***DC*** می باشد و دیگری کنترل کنندهٔ جریان است که برای کنترل جریان موتور ***DC*** می باشد. شکل زیر سیستم کنترل آبشاری موتور ***DC*** را نشان می دهد.

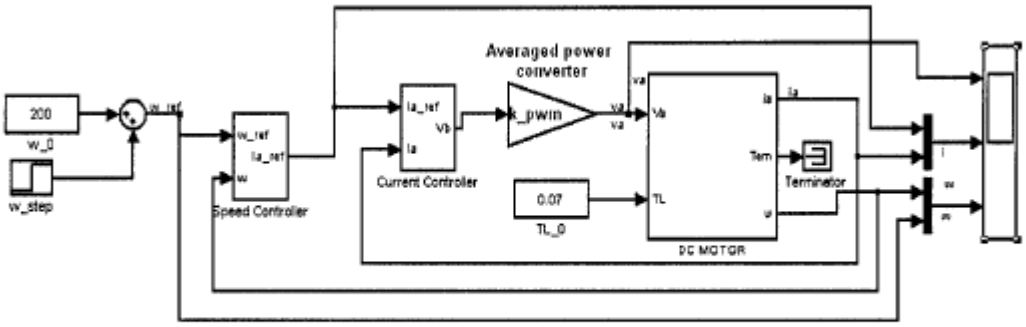

Figure 1.4. Cascade control of DC motor

کنترل کننده سرعت که حلقه خارجی می باشد دو ورودی دارد. یکی از ورودی ها سرعت مطلوب (مرجع)می باشد و ورودی دیگر سرعت موتور ***DC*** می باشد که از خروجی موتور فیدبک گرفته می شود.

کنترل کنندۀ سرعت به گونه ای طراحی می شود که خروجی آن ،سیگنال مطلوب (مرجع) برای کنترل کنندۀ جریان باشد. کنترل کننده سرعت در بلوک دیاگرام فوق حساسیت ها را نسبت به نوسانات بهره کاهش می دهد. حلقۀ داخلی در بلوک دیاگرام کنترل آبشاری ، کنترل کنندۀ جریان می باشد. خطای محاسبه شده جریان مطلوب (مرجع) که از کنترل کنندۀ سرعت گرفته می شود و جریان واقعی که از موتور گرفته می شود و ورودی این کنترل کننده می باشد. این کنترل کننده برای بهتر کردن پاسخ گذاری سیستم و محدود کردن جریان موتور استفاده می شود. خروجی کنترل کنندۀ جریان به یک مدوله کنندۀ پهنای پالس ***PWM*** (***Pulse width modulator***)داده می شود که ولتاژ ترمینال خروجی را کنترل می کند.

سیستم استنتاج فازی – عصبی تطبیقی (***ANFIS***) در فصل *2* بحث می شود. در بحث سیستم استنتاج فازی ، سیستم استنتاج فازی « سوگنو » که در ***ANFIS*** استفاده می شود توضیح داده می شود.

الگوریتم یادگیری هیبریدی که کلید اصلی***ANFIS*** است ، در ادامه بحث یادگیری ***ANFIS*** به آن پرداخته می شود. در فصل سوم راهکارهای طراحی کنترل کننده ارائه می شود. طراحی کنترل کننده***PID*** در قسمت (۱-۳) این فصل بررسی می شود که درابتدا مشخصات کنترل کننده های***D, I, P*** در آن بحث می شود و سپس کنترل کننده های ***PID*** برای طراحی کنترل کننده سرعت و جریان از نوع ***PID*** توضیح داده می شود.

طراحی کنترل کنندۀ ***ANFIS*** در قسمت *(2-3)* از این فصل توضیح داده می شود . نتایج شبیه سازی کنترل کننده های***PID*** و ***ANFIS*** درفصل چهارم ارائه می شود. در فصل عملکرد کنترل کننده های

PID و ***ANFIS*** برای گشتاورهای بار مختلف ارزیابی می شود .اجرای عملی کنترل کننده های ***PID*** و ***ANFIS*** و نتایج آن در فصل پنجم مورد بررسی قرار می گیرد.

فصل دوم

۲. سیستم های استنتاج فازی – عصبی تطبیقی

ANFIS مخفف (***Adptive Network- base Inference System***) یا (***Adptive Neuro- Fuzzy base Inference System***)

به معنای سیستم استنتاج فازی- عصبی تطبیقی می باشد . یک شبکهٔ عصبی تطبیق دارای مزیت های توانایی یادگیری ، بهینه سازی و متعادل سازی می باشد. در حالی که یک منطق فازی یک روش مبتنی بر قواعد می باشد که این قواعد توسط علم افراد خبره ساخته می شوند. عملکرد و تاثیر خوب منطق فازی در سیستم های غیرخطی و پیچیده به اثبات رسیده است.

ANFIS مزیت استفاده از شبکهٔ عصبی تطبیقی و منطق فازی را با هم ترکیب میکند.

۲-۱. سیستم استنتاج فازی

سیستمی که یک نگاشت از ورودی به خروجی را با استفاده از منطق فازی فرموله میکند به نام سیستم استنتاج فازی ***FIS*** (***Fuzzy Inference System***) شناخته می شود. سیستم استنتاج فازی همچنین به نام سیستم مبتنی بر قواعد نیز نامیده می شود .زیرا این سیستم ها از تعدادی عبارت« اگر – آنگاه » ساخته شده است. وقتی چنین سیستم هایی در نقش کنترلی ظاهر می شوندبه آنها کنترل کننده های فازی می گویند .معماری اصلی ***FIS*** از پنج بلوک تابع تشکیل شده که در شکل زیر نشان داده شده است:

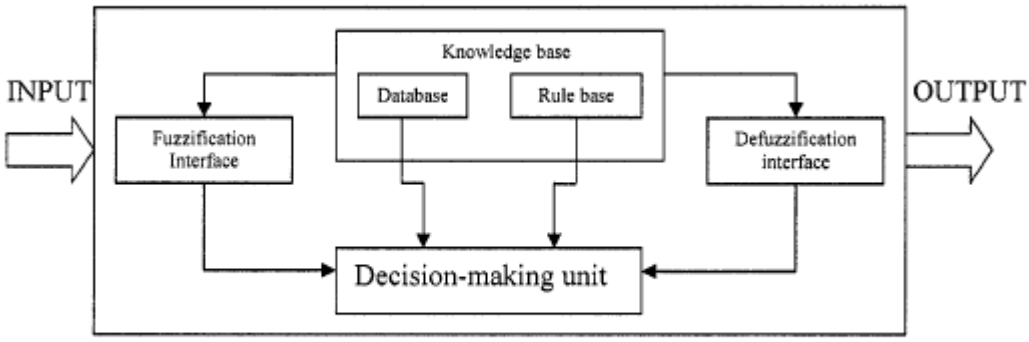

Figure 2.1. Fuzzy Inference System [10]

پایگاه قواعد (***Rule base***) : شامل قواعد و عبارات «اگر – آنگاه » فازی

پایگاه داده ***(Dara base)*** : تعریف توابع عضویت

واحد تصمیم گیری ***(Decision making unit)***: انجام عملیات روی قواعد فازی

رابط فازی ساز ***(Fazzification inter face)***: تبدیل ورودی های حقیقی به مجموعه های فازی

رابط غیر فازی ساز (***Defazzification inter face):*** تبدیل نتایج فازی به مقادیر حقیقی

دو واحد پایگاه داده و پایگاه قواعد با هم تحت عنوان پایگاه دانش ***(Knowledge base)*** شناخته می شوند. سیستم های استنتاج فازی را می توان به سه کلاس « ممدانی ***(mamdani)*** » ، «سوگنو ***(sugeno)*** » و «تاکاگی ***(Takagi)***» تقسیم کرد.

بسیاری از ***FIS*** ها از نوع ممدانی هستند که در این نوع ، اعضای مجموعهٔ فازی خروجی را پیش بینی می کنند. این در حالی است که در نوع سوگنو اعضای مجموعهٔ فازیِ‌خروجی یا رابطهٔ خطی با هم دارند و یا ثابتند.

ANFIS از نوع سوگنو استفاده می کند.

۲-۱-۱. سیستم استنتاج فازی از نوع سوگنو

تاکاگی و سوگنو و کانگ مدل فازی سوگنو را در سال *1985* پیشنهاد و معرفی کردند. در این مدل اعضای مجموعهٔ فازی خروجی یا رابطهٔ خطی با هم دارن یا ثابتند. یک عبارت یا قاعدهٔ فازی نوعی می تواند به صورت زیر باشد:

« اگر ورودی اول برابر ***x*** و ورودی دوم برابر با ***y*** باشد آنگاه خروجی ***z=f(x,y)*** است »

اگر ***f(x,y)*** به صورت یک چندجمله ای درجه *1* باشد، ***FIS*** را مدل فازی از درجه *1* می گویند. اگر ***f(x,y)*** ثابت باشد ***FIS***، مدل فازی سوگنو از درجه صفر نامیده می شود. یک مدل فازی سوگنو در شکل *2.2* نشان داده شده است که در آن خروجی کل از میانگین وزنی همهٔ‌خروجی های حقیقی بدست می آید.

گاهی اوقات به منظور کاهش زمان یادگیریِ ***FIS*** ، به جای میانگین وزنی از مجموع وزن های خروجی استفاده می شود.

۲-۲. شبکه های تطبیقی

یک ساختار شبکه ای که تعدادی گره را بوسیلهٔ تعدادی لینک بـه هـم مربـوط مـی سـازد، یـک شـبکهٔ تطبیقی را تعریف میکند . گره ها در واقع واحدهای پردازش را بیان می کنند. ولینکهـا در واقـع اتصـال بین آن واحدهای پردازشی را بیان می کند. همهٔ گره ها یا حداقل قسـمتی از آنهـا منطبـق بـر طبیعـت هستند یعنی خروجی سیستم از روی پارامترهای گره سـاخته مـی شـود .قواعـد یـادگیری بـه گونـه ای ساخته می شوند که خطای سیستم را کم کنند و پارامترهای گره را آن گونه که باید باشند ،اصلاح کند.

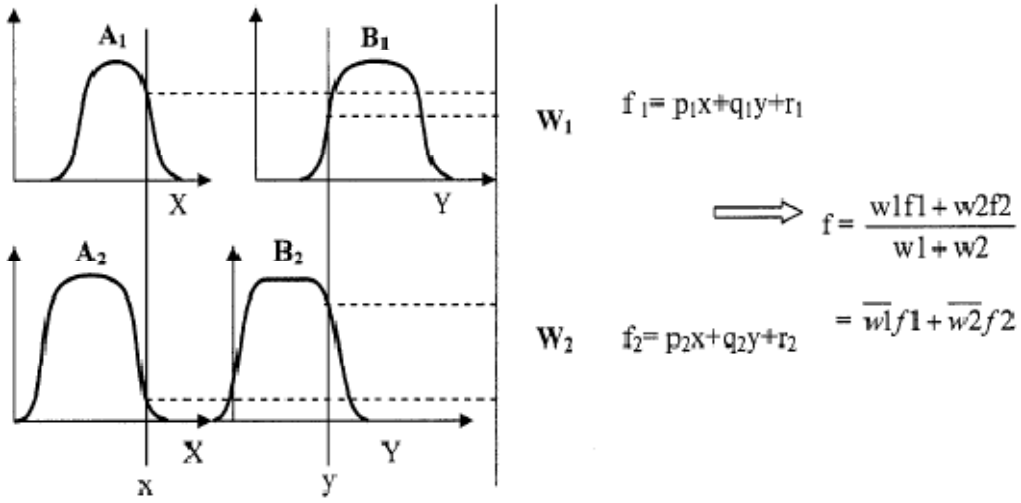

Figure 2.2. First order Sugeno fuzzy model [11]

روش « تندترین شیب » (***Steepest descent***)که توسط « وِربو» معرفی شد . قاعده یـادگیری پایـه ای است که چگونگی بدست آوردن بردار گرادیان را توضیح می دهد.

بردار گرادیان مشتق خطای اندازه گیری شده نسبت به پارامترها می باشد.

یک نمونه ساده روش تندترین شیب به نام قاعده یادگیری ***back- propagation*** شناخته می شود کـه در آن بردار گرادیان در جهت عکس خروجی محاسبه می شود . خطا به صورت اختلاف بین بـردار مرجـع و خروجی واقعی تعریف می شود.

$$E_p = \sum_{k=1}^{N(L)} (d_k - x_{L,k})^2 \qquad (2.1)$$

که در آن ***dk*** . ***k*** امین مجموعهٔ خروجی مطلوب و ***XL,K*** خروجی واقعی و ***L*** تعداد لایه ها می باشد.

به طور کلی دو نوع الگوی یادگیری برای شبکه های تطبیقی ممکن است:

الف- یادگیری آفلاین : در این نوع یادگیری پارامترها ، فقط زمانی که اطلاعات خام به سیستم ارائه مـی شوند ،آبدیت می شوند . این نوع یادگیری را یادگیری گروهی می گویند.

ب- یادگیری آنلاین : در این نوع یادگیری پارامترها ، بعد از اینکه یک ورودی یا خروجی یا هر دوی آنهـا ارائه شدند ،سریعاً آپدیت می شوند.

همچنین می توان دو روش فوق را با هم ترکیب کـرد و پارامترهـا را بعـد از ***k*** عضـو مجموعـه دیتاهـای یادگیری آپدیت کرد که ***k***بیانگر مقدار (***epoch***) می باشد.

ANFIS از قاعده یادگیری هیبریدی که از ترکیب روش گرادیان و روش حداقل مربعـات مـی باشـدبرای تعیین پارامترها استفاده می کند. الگوریتم یادگیری هیبریدی تنها زمانی که خروجـی ترکیـب خطـی از ورودی باشد قابل اجرا می باشد.

۲-۳. ساختار *ANFIS*

یک مدل فازی سوگنو درجه یک با دو ورودی x و y و یک خروجی z در نظر بگیرید . قواعـد فـازی نوعـاً می توانند به صورت زیر باشند:

قاعده *1*: اگر $x=A1$ و $y=B1$ آنگاه $f1=p1x+q1y+r1$

قاعده *2*: اگر $x=A2$ و $y=B2$ آنگاه $f2=p2x+q2y+r2$

خروجی کل میانگین وزنی خروجی ها می باشد . ساختار ***ANFIS*** عبارات فوق در شکل *2.3* نشـان داده شده است:

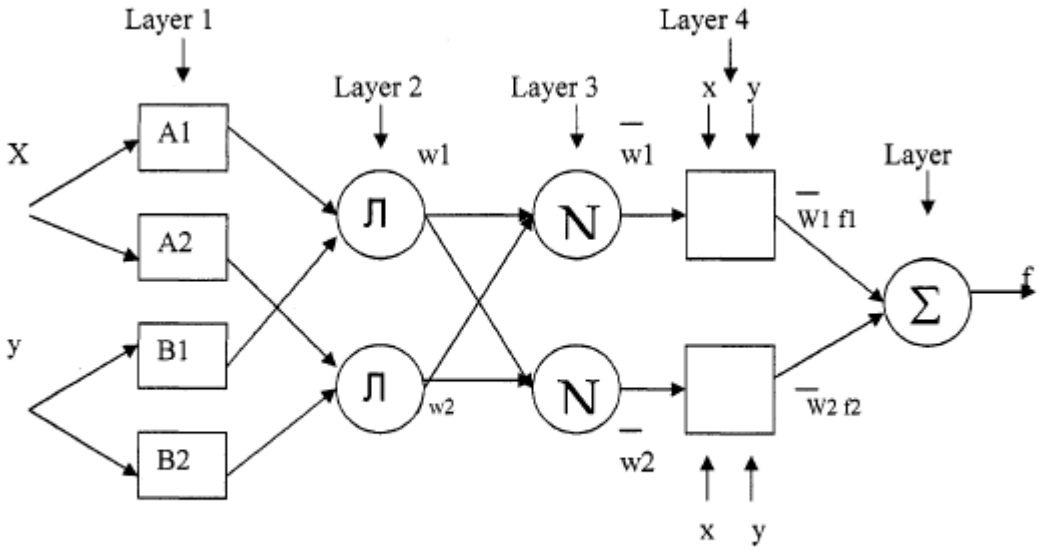

Figure 2.3. ANFIS Architecture [11]

این ساختار پنج لایه دارد .ملاحظه می شود گره های لایه های یکسان ، توابع یکسان دارند. گره خروجیِ ***I*** در لایهٔ ***l*** با ***Ol,i*** نامگذاری می شود.

لایهٔ *1*: هر گره در این لایه شامل یک گره تطبیقی با یک تابع گره است و داریم :

$$O_{1,i} = \mu A_i(x) \text{ for } i = 1,2 \text{ or, } O_{1,i} = \mu B_{i-2}(y), \text{ for } i = 3,4$$

که در آن ***x*** (یا ***y***) ورودی گره ***I*** و ***Ai*** (یا ***Bi-2***) یک مجموعهٔ فازی مربوط به آن گره می باشد. به عبارات دیگر خروجی این لایه مقدار عضویت آن می باشد.

توابع عضویت برای ***A*** می توانند هر تابع عضویت پارامتری متناسب باشند.

هر پارامتر در این لایه به عنوان یک پارامتر پیش فرض نام برده می شود.

لایهٔ *2* : هر گره دراین لایه با ***n*** نامگذاری شده است و خروجی هرگره حاصلضرب همهٔ سیگنالهای ورودی به آن گره می باشد. این گره ها عمل ***AND*** فازی را انجام می دهند و داریم :

$$O_{2,i} = \mu A_i(x) \, X \, \mu B_i(y) \text{ for } i = 1,2$$

که در آن خروجی هر گره قدرت آتش هر قاعده ***(firing strenght)*** را نشان می دهد.

لایهٔ *3* : هر گره در این لایه با ***N*** نامگذاری شده است .گره های این لایه خروجی نرمالیزه شده هر قاعده رامحاسبه می کند و داریم :

$$O_{3,i} = \frac{\overline{wi}}{\sum_i wi}, \quad i=1,2,$$

که در آن ***wi*** قدرت آتش ***(firing strenght)*** آن قاعده می باشد. خروجی این لایه قدرت آتش نرمالیزه شده نامیده می شود.

لایهٔ *4*: هرگره در این لایه وابسته به یک تابع گره می باشد و داریم:

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \quad i = 1, 2,$$

که در آن ***wi*** ، قدرت آتش نرمالیزه شده از لایهٔ سوم است و***{pi , qi , ri}*** مجموعهٔ پارامترهای گره ***i*** هستند . پارامترهای این لایه تحت عنوان « پارامترهای نتیجه شده » نامیده می شوند.

لایهٔ *5*: تنها گره موجود در این لایه با ***Σ*** نامگذاری شده است که مجموع تمام سیگنالهای ورودی به آن را محاسبه کرده و به خروجی می برد. لذا داریم:

$$O_{5,i} = \sum \overline{wi}\, fi = \frac{\sum_i wi\, fi}{\sum_i wi} \qquad i = 1, 2$$

که در آن ***O5,i*** خروجی ***i*** اُمین گره در لایهٔ پنجم می باشد.

۲-۴. الگوریتم یادگیری هیبریدی

الگوریتم یادگیری هیبریدی ترکیبی از روش های حداقل مربعات و تندتری شیب می باشد. خروجی کل ***f*** از شکل ۲.۳ می تواند به صورت زیر بیان شود.

$$f = \frac{w1}{w1+w2} f1 + \frac{w1}{w1+w2} f2$$

$$= \overline{w1}\, f1 + \overline{w2}\, f2$$

$$= (\overline{w1}\, x)p1 + (\overline{w1}\, y)q1 + (\overline{w1})r1 + (\overline{w2}\, x)p2 + (\overline{w2}\, y)q2 + (\overline{w2})r2 \qquad (2.2)$$

از معادلات فوق می توان این را بیان داشت که خروجی ترکیبی خطی از پارامترهای نتیجه شدهٔ ***r2 , q2,*** ***p2 , r1, q1, p1*** می باشد. بنابراین وقتی پارامترهای پیش فرض ثابت هستند، خروجی کل ترکیبی خطی از پارامترهای نتیجه شده است و در این صورت الگوریتم یادگیری هیبریدی عملی می شود. جدول زیر الگوریتم یادگیری هیبریدی را کاملاً بیان می کند.

	Forward Pass	Backward Pass
Premise Parameters	Fixed	Gradient Descent
Consequent Parameters	Least-Squares Estimate	Fixed
Signals	Node Outputs	Error Signals

Table 2.1. Two passes in the hybrid-learning algorithm [11]

در مرحلهٔ بعد خروجی گره ها به لایهٔ چهارم می رسند و پارامترهای نتیجه شده با تخمین حداقل مربعات بیان می شوند در حالیکه در مرحلهٔ قبل پارامترهای پیش فرض با روش گرادیان آبدیت می شدند.

اگر پارامترهای پیش فرض ثابت باشند ، پارامترهای نتیجه شده که در مرحلهٔ بعد بدست می آیند بهینه هستند.بنابراین روش یادگیری هیبریدی خیلی سریع تر از روش ***back – propagation*** است.

۲-۴-۱. نکات

برای افزایش عملکردسیستم می توانیم از ترکیب چندالگوریتم تندترین- شیب و حداقل مربعات استفاده کنیم .

توصیه می شود الگوریتم های یادگیری ماشین روی توابع حقیقی صورت نگیرد. بلکه ابتدا آنها را به صورت عبارات فازی بیان کنیم و سپس از الگوریتم های یادگیری استفاده کنیم.

اگر تعداد دیتاهای یادگیری زیاد باشند ، توابع عضویت می توانند با دقت بسیار زیادی تنظیم شوند.

۲-۵. محدودیت های *ANFIS*

۱-***ANFIS*** فقط سیستم های سوگنو درجه صفر و درجه یک را پشتیبانی می کند.

۲- همهٔ توابع عضویت یا ترکیب خطی ازهم می باشند و یا ثابتند.

۳- تعداد توابع عضویت باید با تعداد قواعد فازی برابر باشند.

۴- وزن هر قاعده فازی برابر واحد است.

۲-۶. یادگیری *ANFIS*

ANFIS می تواند با الگوریتم یادگیری هیبریدی یاد بگیرند. پارامترهای پیش فرض که توابع عضویت را تعریف می کنند با استفاده از روش گرادیان تعیین می شوند و پارامترهای نتیجه شده یا به کار بردن روش حداقل مربعات تعیین می شوند .

ANFIS می تواندر ***MATLAB*** از ***command line*** و یا یک ***GUI*** مربوط به ***ANFIS*** در ***MATLAB*** به نام ***ANFIS Editor*** به کار گرفته شود.

ANFIS Editor GUI : با تایپ کردن ***anfis edit*** در ***command line*** نرم افزار ***MATLAB*** پنجرهٔ ***ANFIS Editor*** باز می شود . شکل ۲.۴ صفحهٔ ***ANFIS Editor*** را با شرح توابع آن نشان میدهد.

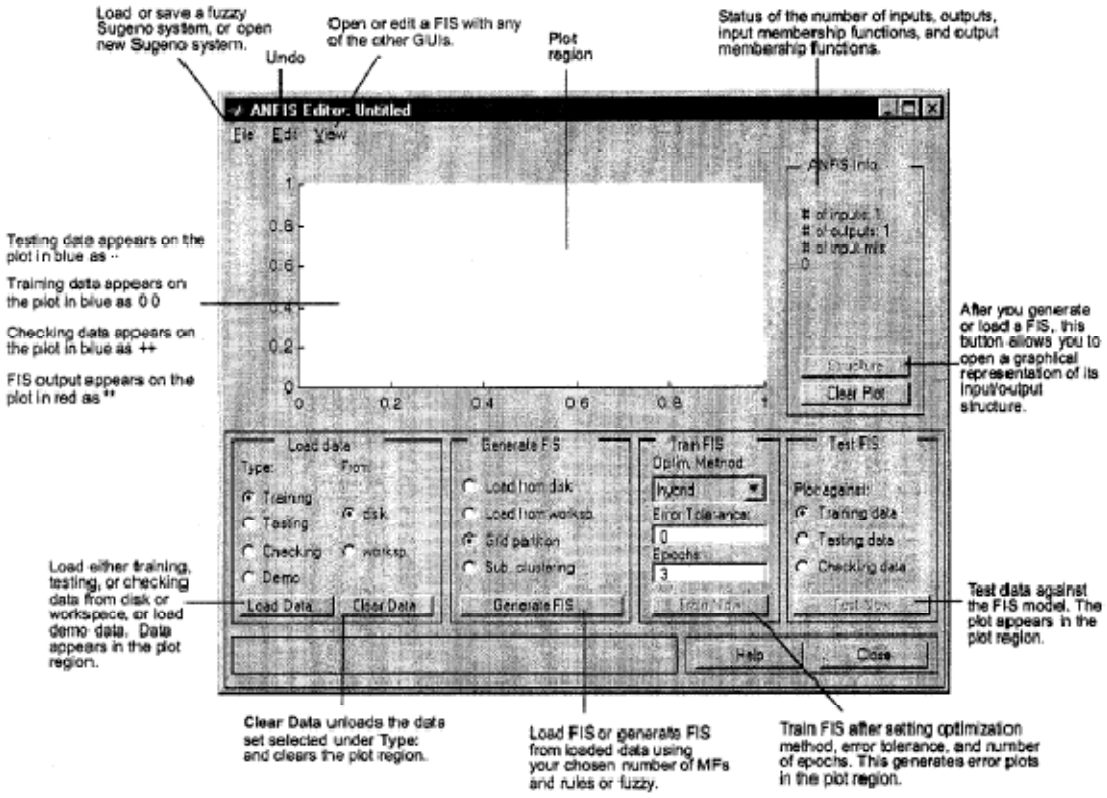

Figure 2.4. ANFIS editor GUI [9]

ANFIS Editor چهار قسمت اصلی دارد که عبارتند از :

Load Data -۱

Generate FIS -۲

Train FIS -۳

Test FIS -۴

با استفاده از ***Load Data*** ما می توانیم***Testing date*** ، ***Checking date*** و ***Traning date*** و یا ترکیبی از آنها را در برنامه ***Load*** کنیم. یکبار که هر کدام از دیتاهای یاد شده در برنامه ***Load*** شوند در ***Plot region*** نمایش داده میشود .وقتی دیتاها ***Load*** شدند می توان یک ***FIS*** بوسیلهٔ « ***Generate FIS*** » تولید کرد. در تولید ***FIS*** نوع و تعداد «توابع عضویت » قابل انتخاب است. ساختار ***FIS*** تولید شده با کلیک کردن روی دکمهٔ « ***Structure***» قابل مشاهده است که در شکل نشان داده شده است:

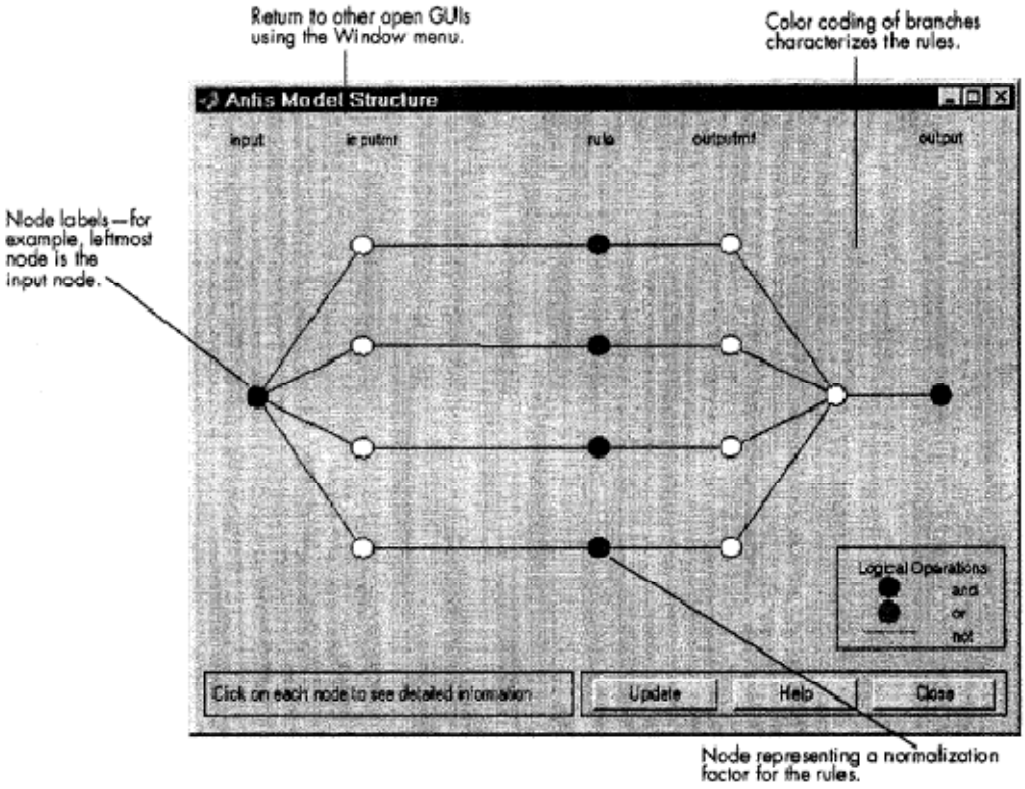

Figure 2.5. Generated FIS Structure [9]

وقتی ***FIS*** اولیه تولید شد،این ***FIS*** می تواند با هر کدام از الگوریتم های یادگیری ***back – propagation*** یا هیبریدی یاد بگیرید. از لحاظ تئوری تلورانس خطا برابر صفر است اما در عمل چیزی حدود *0/01* خواهد بود. خطای یادگیری در عمل بیشتر خواهد بود اما با افزایش توابع عضویت می توان عملکر سیستم بهتر و خطای یادگیری را کمتر کرد. هر وقت که یادگیری ***FIS*** تمام شد آن را در برابر ***Checking date*** و ***Testing date*** امتحان می کنند تا میزان یادگیری ***FIS*** آزمایش شود.

فصل سوم

۳. طراحی کنترل کننده

کنترل کننده ها نقش حیاتی را در کنترل سرعت موتور ***DC*** به میزان دلخواه بازی می کنند. در فصل های قبل دو نوع کنترل کننده معرفی شد وعملکرد آنها ارزیابی شد.یکی از آنها کنترل کنندهٔ ***PID*** (***Proportional - Integral – Derivative***) و دیگری که یک کنترل کنندهٔ هوشمند بود به نام ***ANFIS*** که عملکرد آنها در گشتاور بارهای مختلف ارزیابی خواهد شد.

۳-۱. طراحی کنترل کننده *PID*

بسیاری از کنترل کننده هایی که در حال حاضر در صنعت هستند ، از نوع ***PID*** می باشند. ورودی ***PID*** یک سیگنال خطا *(* اختلاف بین سیگنال مرجع و سیگنال خروجی *)* می باشد. خروجی حاصل جمع سه جمله می باشد . جملهٔ اول متناسب با سیگنال خطا، جملهٔ دوم متناسب با انتگرال سیگنال خطا و جملهٔ سوم متناسب با مشتق سیگنال خطا می باشد. تابع تبدیل ***PID*** به صورت زیر بیان می شود.

$$u(t) = ke(t) + k_i \int_0^t e(\tau)d\tau + k_d \frac{de}{dt} \qquad (3.1)$$

که در آن :

(***KP***) بهرهٔ متناسب = ***K***

بهرهٔ انتگرال = ***Ki***

بهرهٔ مشتق = ***Kd***

سیگنال خطا = ***e(t)***

معادلهٔ *(1-3)* را می توان به صورت زیر نوشت :

$$u(t) = k(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d \frac{de(t)}{dt}) \qquad (3.2)$$

که در آن ***Ti*** بازهٔ زمانی انتگرال گیری و ***Td*** بازهٔ زمانی مشتق گیری می باشد. پارامترهای دو معادلهٔ *1.3* و *3.2* به صورت زیر به هم مربوط می شوند :

$$k = K \qquad k_i = \frac{K}{T_i} \qquad k_d = KT_d \qquad (3.3)$$

۳-۱-۱. مشخصات کنترل کنندهٔ PID

کنترل کننده متناسب ***(Proportional)*** زمان خیز (***rise time***)سیستم را کاهش می دهد اما این کنترل کنندهٔ خطای حالت ماندگار را حذف نمی کند . کنترل کنندهٔ انتگرال گیر می تواند خطای حالت ماندگار را کاهش دهد اما اُورشوت را افزایش می دهد. کنترل کنندهٔ مشتق گیر پایداری سیستم را افزایش می دهد و همچنین اورشوت را کاهش می دهد. عملکرد هر کدام از سه کنترل کنندهٔ فوق در جدول زیر خلاصه شده است :

Controller	Rise time	Overshoot	Settling time	Steady-state error
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Small Change	Decrease	Decrease	Small Change

Table 3.1. Characteristics of each control parameters [14].

۳-۱-۲. تنظیم کردن PID

کنترل کنندهٔ ***PID*** به چند روش می تواند تنظیم شود. به غیر از روش های عمومی که برای تنظیم ***PID*** به کار می رود . چند روش خاص وجود دارد که توسط « زیگلر » و « نیکول » معرفی شدند که با تعداد پارامتر کم و معادلات آسان در ارتباط است .یکی از روش ها بر پاسخ شیب پایه گذاری شده است. روش دیگر روش پاسخ فرکانسی است.کنترل کنندهٔ ***PID*** که در این پایان نامه بحث شده با روش پاسخ فرکانسی طراحی شده است .این روش بر پیدا کردن نقطه تلاقی نمودار نایکوئیست با قسمت منفی محور حقیقی مبتنی است.

پارامترهای ***Ku*** (بهره نهایی) و ***Tu*** (پریود نهایی) از روی نمودار نایکوئیست تعیین می شوند. جدول *2.3* جهت تعیین پارامترهای ***PID*** به ما کمک می کند. یکبار که پارامترهای ***D, I, P*** تعیین شدند این پارامترها بوسیله روش آزمایش و خطا به گونه ای تنظیم می شوند که خروجی مطلوب بدست آید.

Controller	K	Ti	Td
P	0.5Ku		
PI	0.4Ku	0.8Tu	
PID	0.6Ku	0.5Tu	0.125Tu

Table 3.2. Controller parameters using Ziegler-Nichols frequency method [15]

۳-۱-۳. کنترل کنندهٔ سرعت *PID*

خطای بین سیگنال مرجع و خروجی واقعی به ورودی کنترل کنندهٔ سرعت داده می شود. یک ورودی شیب بین *20۰* تا *400* به عنوان مرجع به سیستم داده می شود.

شکل *1.3* : دیاگرام یک کنترل کنندهٔ سرعت را که در سیمولینک کشیده شده نشان می دهد.

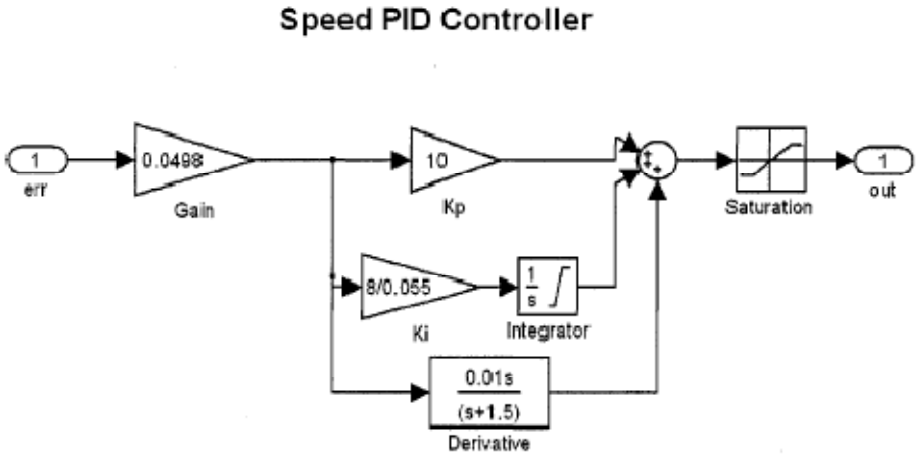

Figure 3.1. Speed PID controller

در زیر پارامترهای تنظیم شده کنترل کنندهٔ سرعت نشان داده شده است .مقدار اشباع روی *5±* تنظیم شده است

$$Kp = 0.498 \qquad Ki = 7.24 \qquad Kd = 0.000488$$

این کنترل کننده به گونه ای طراحی می شود که خروجی آن به عنوان سیگنال مرجع برای کنترل کنندهٔ جریان ***PID*** باشد.

۳-۱-۴. کنترل کننده جریان*PID*

ورودی کنترل کننده جریان ***PID*** یک سیگنال خطا است. این سیگنال خطا اختلاف بین جریان مرجع *(*که از خروجی کنترل کنندهٔ سرعت گرفته می شود *)* و جریان واقعی *(*که همان جریان مصرفی موتور است*)* می باشد. مقدار اشباع روی ± 1 تنظیم شده است. در زیر پارامترهای ***D, I, P*** از کنترل کنندهٔجریان ***PID*** نشان داده شده است .

$$Kp = 0.04 \qquad Ki = 0.366 \qquad Kd = 0.000412$$

شکل زیر یک کنترل کننده جریان ***PID*** را نشان می دهد.

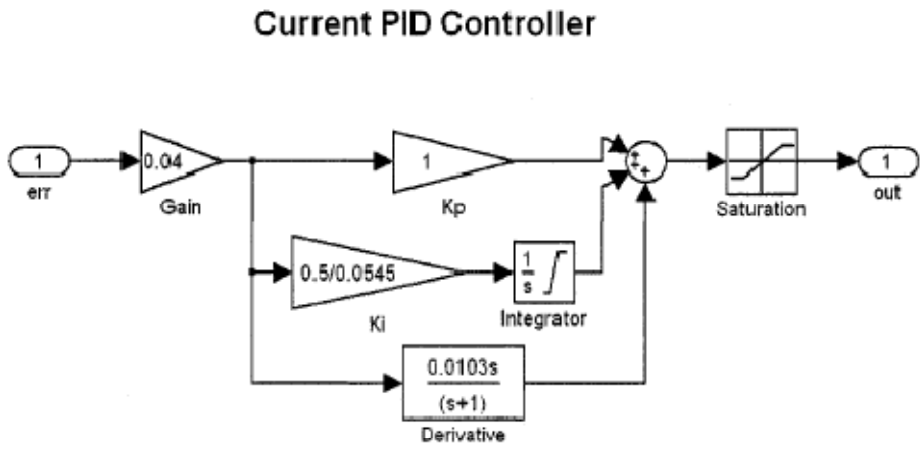

Figure 3.2. Current PID controller

خروجی کنترل کننده جریان ***PID*** به یک مبدل توان داده می شود که برای مدوله کردن پهنای پالس ولتاژ در کنترل کنندهٔ توان استفاده می شود . خروجی کنترل کننده توان متوسط به عنوان تغذیه به موتور ***DC*** داده می شود.

۳-۲. طراحی کنترل کنندهٔ *ANFIS*

همانطور که در فصل *2* بحث شد دیتاهای آموزشی ***(Traning date)*** برای طراحی ***ANFIS*** حتماً نیاز است. دیتاهای چک کننده ***(Checking date)*** نیز برای چک کردن سیستم از لحاظ اینکه یادگیری سیستم درست صورت گرفته یا نه مورد نیاز می باشند. دو نوع کنترل کنندهٔ ***ANFIS*** در این فصل طراحی می شوند. یکی برای کنترل سرعت و دیگری برای کنترل جریان. شبیه سازی موتور ***DC*** به همراه کنترل کنندهٔ ***PID*** برای تولید دیتاهای آموزش استفاده می شود. دیتاهای آموزشی اینجا چیزی جز ورودی ها و خروجی های کنترل کنندهٔ ***PID*** نیست.

در مرحلهٔ اول ***FIS*** مربوط به سرعت با استفاده از دیتاهای آموزشی جمع آوری شده از کنترل کننده ***PID*** ساخته می شود . قسمتی از دیتاهای آموزشی به عنوان دیتاهای چک کننده کنار گذاشته می شود تا بعد از پایان یادگیری کنترل کننده ، توسط آنها یادگیری کنترل کننده چک شود.

شکل *3.3* ، پنجرهٔ یک ***ANFIS Editor*** را نشان می دهد که در آن دیتاهای آموزشی و دیتاهای چک کنندهٔ نمایش داده شده است.

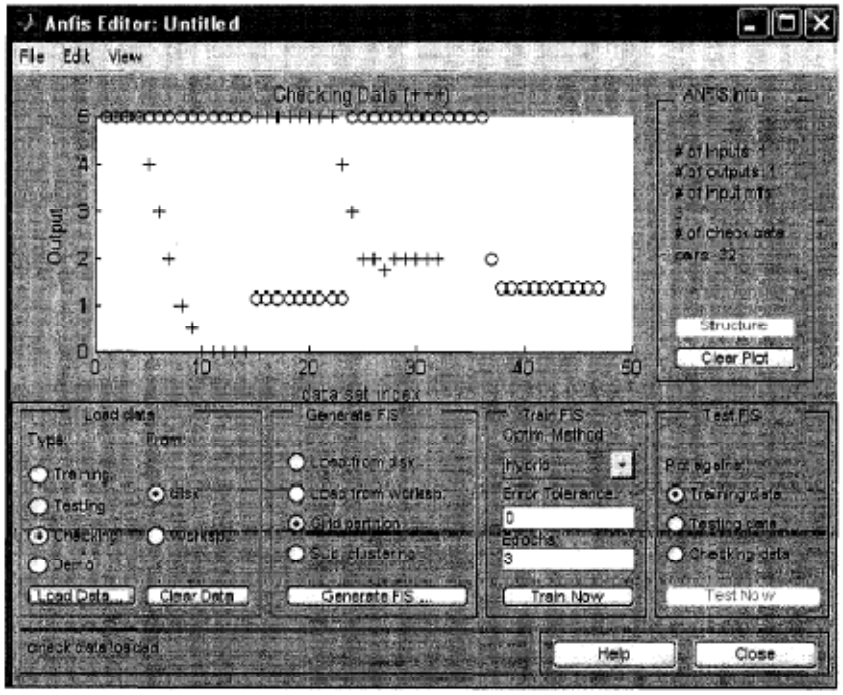

Figure 3.3. ANFIS GUI after training and checking data is loaded.

دیتاهای آموزشی و چک کننده در یک فایل دیتا ذخیره می شوند.

با تایپ کردن ***anfis edit*** در ***MATLAB command*** پنجرهٔ ***ANFIS Editor***باز میشود.دیتاهای آموزشی و چک کننده با ***Load*** کردن فایل مربوطه در برنامه ***load*** می شوند. یکبار که دیتاها در برنامه ***Load*** شدند، ***GUI*** ظاهر میشود ،همانطور که در شکل دیده می شود.

سمبل « ***O***»معرف دیتاهای آموزش و سمبل « +» معرف دیتاهای چک کننده می باشند. ***FIS*** اولیه با کلیک کردن روی « ***Generate FIS*** » در حالی که تنظیمات روی حالت پیش فرضِ « ***Grid partition*** » است، ساخته می شود . با این کار پنجره ای شبیه شکل ۳.۴ باز می شود:

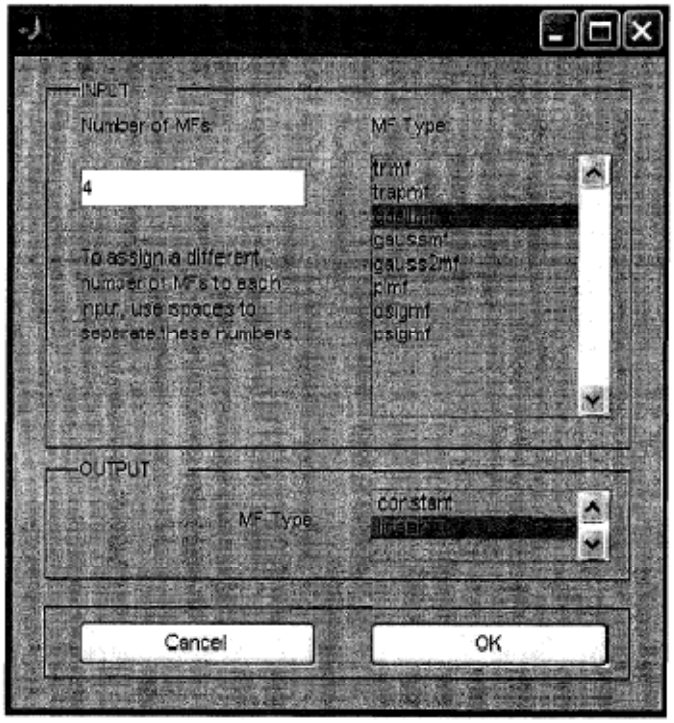

Figure 3.4. GUI showing selection of I/O membership functions

ملاحظه می شود که توابع عضویت ***MF***(***Membership Function***) روی گزینهٔ ***gbellmf*** و تعداد توابع عضویت روی *4* انتخاب شده است. همانطور که در شکل فوق دیده می شود برای توابع عضویت خروجی دو انتخاب وجود دارد .

توابع عضویت یا می توانند ثابت انتخاب شوند یا خطی . همان چیزی که یک ***FIS*** از نوع سوگنو احتیاج دارد.

وقتی یک ***FIS*** ساخته می شود آماده هستیم تا بوسیلهٔ دیتاهای آموزشی آن را تحت آموزش قرار دهیم. در اینجا ما تلورانس را روی *0/01* تنظیم می کنیم. تعداد تکرار لازم از روی (***epoch***) معلوم می شود. تعداد تکرار پیش فرض *3* می باشد که برای یادگیری مناسب نیست .بنابراین تعداد تکرار به صورت رندوم روی *50* تنظیم می شود . اکنون سیستم آماده یادگیری با استفاده از الگوریتم یادگیری هیبریدی است. با کلیک کردن روی دکمهٔ «***Train now***» ، ***FIS*** شروع به یادگرفتن می کند . اضافه کردن توابع عضویت بیشتر عملکرد ***FIS*** تولید شده را بهبود می بخشد.

عملکرد ***FIS*** در برابر دیتاهای چک کننده تست می شود . تصویر زیر نتایج را بعد از اینکه تست انجام شد نشان می دهد:

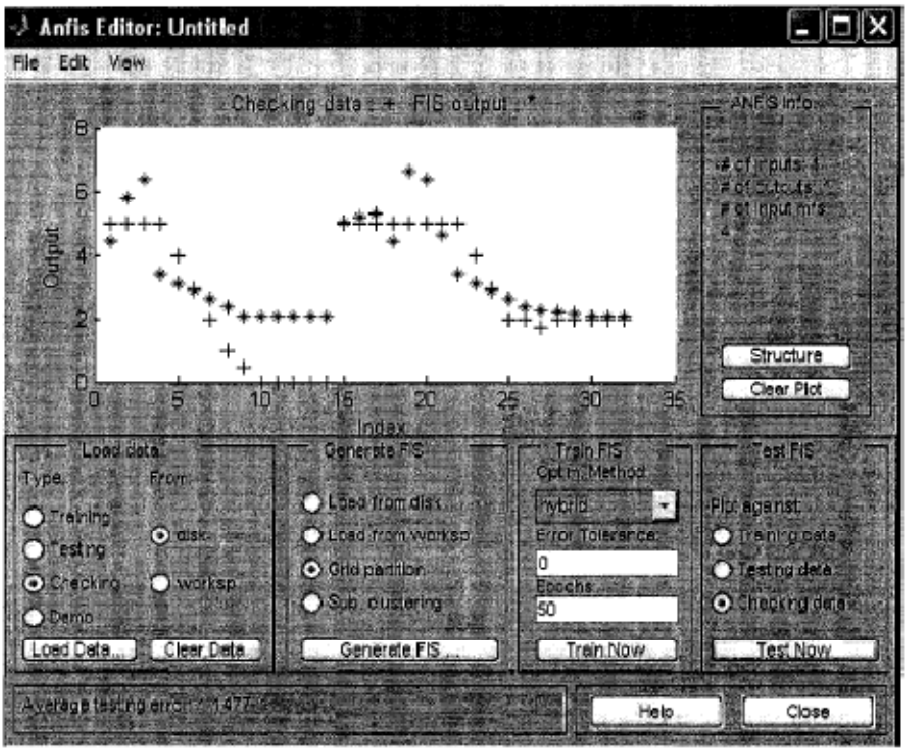

Figure 3.5. Screen after testing the FIS is done.

خروجی با سمبل « * » و ورودی با سمبل « + » نشان داده شده است. ساختار ***FIS*** تولید شده در شکل زیر نشان داده شده است.

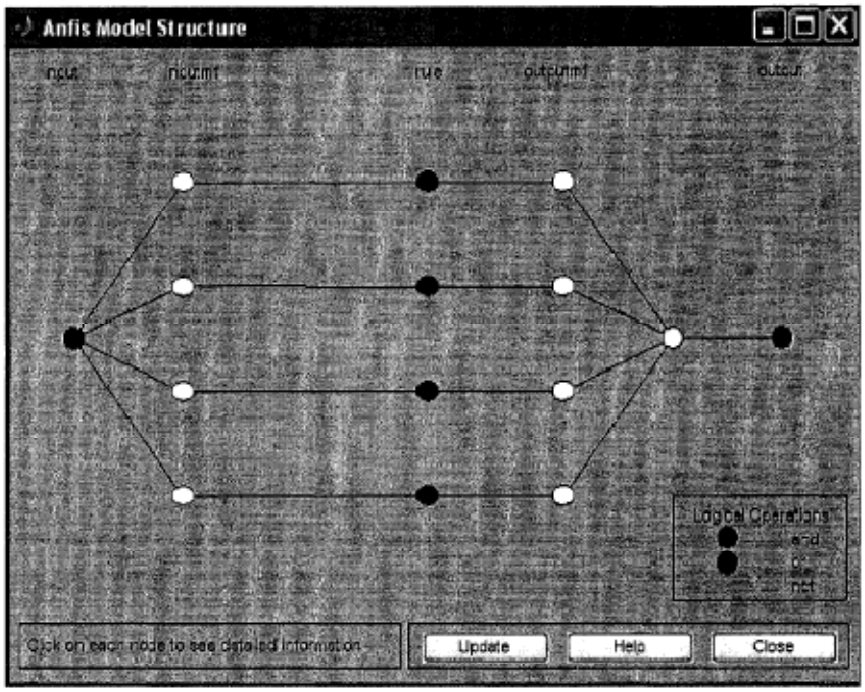

Figure 3.6. Structure of the generated FIS for speed controller.

به همین نحو ***FIS*** مربوط به جریان نیز ساخته می شود.

به وسیلهٔ ***fuzzy block*** در ***simulink liberary*** یک مدل برای کنترل کنندهٔ ***ANFIS*** برای موتور ***DC*** طراحی می شود . در شکل زیر بعضی از بلوکها نوشته شده ***Gain*** که این ها بلوکها بهره می باشند. بهره ها طوری تنظیم می شوندکه خروجی مناسب تولید شود.

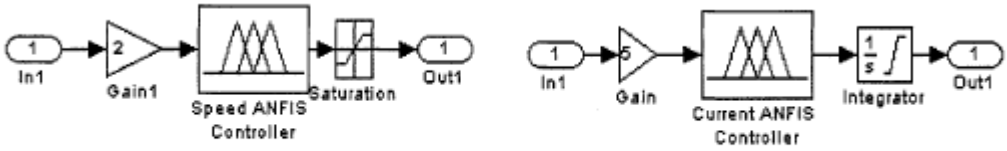

Figure 3.7. Speed and Current ANFIS controllers

بلوک دیاگرام مربوط به ***ANFIS*** شبیه به بلوک دیاگرام ***PID*** است با این تفاوت که در آن ***PID*** مربوط به سرعت و جریان به ترتیب با***ANFIS*** مربوط به سرعت و جریان جایگزین شده اند. برای هر کدام از ***ANFIS*** های مربوط به جریان و سرعت ورودی سیستم یک سیگنال خطاست که اختلاف بین سیگنال مرجع و سیگنال خروجی می باشد . بلوک دیاگرام های ***ANFIS*** سرعت و ***ANFIS*** جریان در شکل *(3.7)* نشان داده شده است.

فصل چهارم

۴. شبیه سازی

وقتی کنترل کننده های سرعت و جریان طراحی شدند و موتور ***DC*** در سیمولینک شبیه سازی شد ، سیستم برای شبیه سازی آماده شده است. سرعت مرجع به صورت یک ورودی شیب به سیستم داده میشود که در آن شیب به صورت ۲۰۰ ***rad/ sec*** در ***t*** = ۰***sec*** و ۴۰۰ ***rad/ sec*** در ***t*** = ۵ ***sec*** تغییر می کند. پارامترهای موتور ***DC*** که قبلاً در یک ***m-file*** (فایل های مربوط به ***MATLAB***) ذخیره شده قبل از شبیه سازی ***Load*** شده و اجرا می شود.این فصل نتایج شبیه سازی کنترل کنندۀ ***PID*** و ***ANFIS*** را بررسی می کند . نتایج شبیه سازی هر دو کنترل کننده برای گشتاور بارهای مختلف مقایسه و ارزیابی می شود.

۴-۱. شبیه سازی ***PID***

کنترل کنندۀ ***PID*** سرعت و ***PID*** جریان که در فصل سوم طراحی شدند در این فصل شبیه سازی می شوند. شکل زیر مدل سمبولیک کنترل کنندۀ ***PID*** را نشان می دهد.

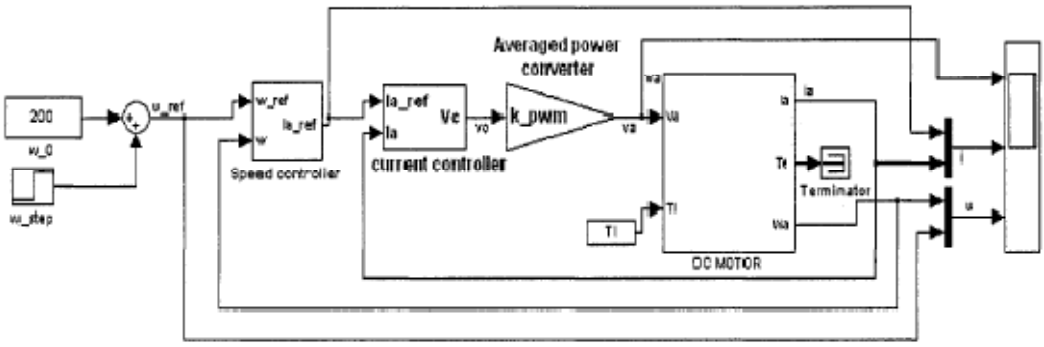

Figure 4.1. Simulation model

نتایج شبیه سازی کنترل کننده ***PID*** در شکل ۴.*2* نشان داده شده است . در این شکل نمودار اول تغییرات ولتاژ ورودی را نشان می دهد . نمودار دوم مربوط به تغییرات جریان و نمودار سوم تغییرات سرعت را نشان میدهد.

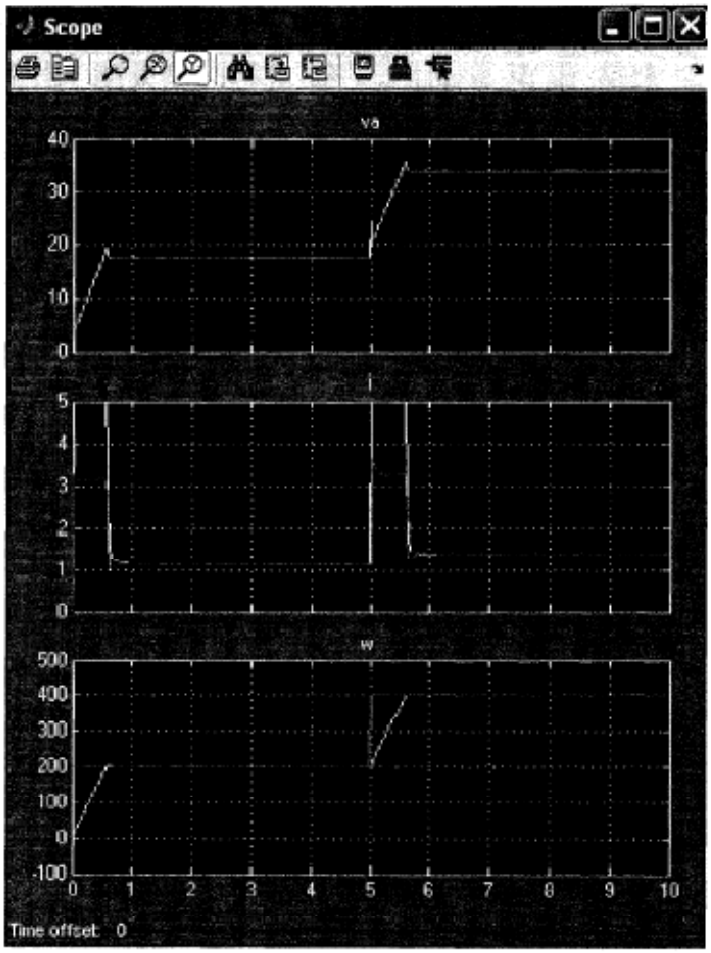

Figure 4.2. Simulation results of using speed and current PID controllers.

جدول ۴.۱ آنالیز تغییر فاکتورهای کنترلی در کنترل کننده موتور ***DC*** از نوع ***PID*** را نشان می دهد.

۴-۲. شبیه سازی *ANFIS*

موضوع این بحث شبیه سازی کنترل کنندهٔ موتور ***DC*** از نوع ***ANFIS*** می باشد. مدل سمبولیک کنترل کنندهٔ ***ANFIS*** شبیه شکل ۴.۱ است که در آن کنترل کننده های ***PID*** جای خود را به کنترل کنندهٔ ***ANFIS*** داده اند.

Factor	Value
Rise time	0.6 sec
Overshoot when speed = 200 rad/sec	203
Settling time	0.9 sec
Steady state error	0

Table 4.1. PID controller analysis

نتایج استفاده از کنترل کنندهٔ ***ANFIS*** درشکل ۴.۲ نشان داده شده است.

جدول ۴.۲ فاکتورهای کنترلی ناشی از شبیه سازی کنترلر ***ANFIS*** را نشان می دهد.

Factor	Value
Rise time	0.4 sec
Overshoot when speed = 200 rad/sec	203
Settling time	0.48 sec
Steady state error	~1.5 rad/sec

Table 4.2. ANFIS controller analysis

شکل ۴.۳ نتایج شبیه سازی کنترل کننده سرعت و جریان ازنوع ***ANFIS*** را نشان می دهد.

۴-۲-۱. مقایسه کنترل کنند های *PID* و *ANFIS* در گشتاور بار ثابت

عملکرد کنترل کنند های ***PID*** و***ANFIS*** در گشتاور بارهای مختلف در جدول هـای *4.1* و *4.2* مـورد بررسی قرار گرفت. چیزی که از مشاهده این جداول بدست می آید این است که کنترلـر ***ANFIS*** نتـایج بهتری نسبت به ***PID*** داشته است . با وجود اینکه خطای حالت ماندگار ***PID*** از ***ANFIS*** کمتر است ولـی روی هم رفته ***ANFIS*** نسبت به ***PID*** عملکرد خیلی بهتری دارد .

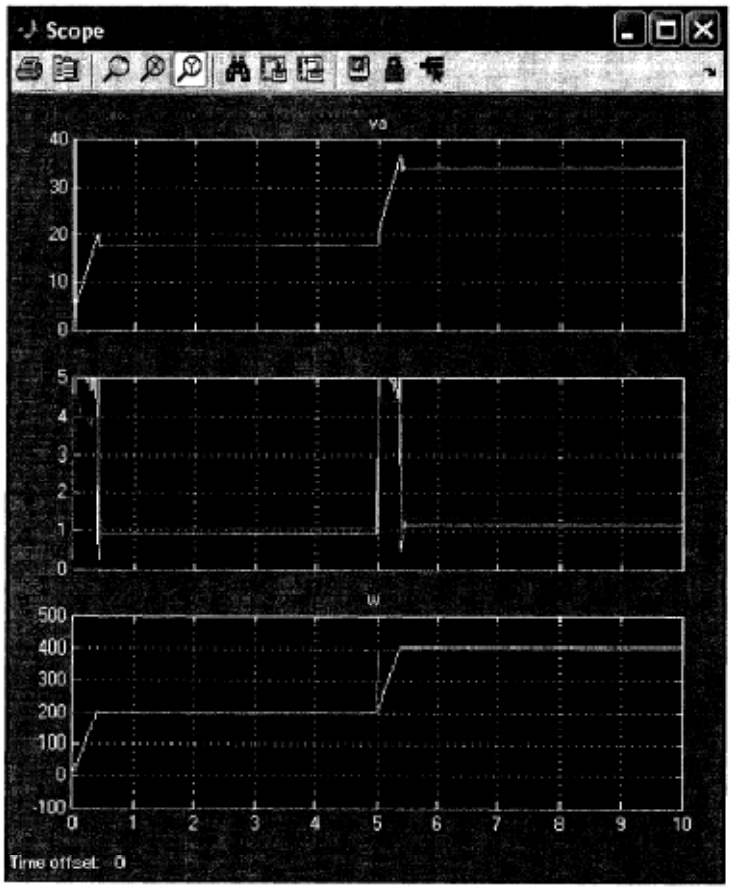

Figure 4.3. Results of using an ANFIS controller.

۴-۲-۲. مقایسه کنترل کنند های *PID* و *ANFIS* در گشتاور بارهای مختلف

در این قسمت عملکرد کنترل کنند های ***PID*** و ***ANFIS*** در گشتاور بارهای مختلف بررسی می شود.

برای این کار ما سه حالت را بررسی می کنیم:

وقتی گشتاور بار ***N-m*** ۰.۱ باشد:

شکل *4.4* نتایج شبیه سازی استفاده از کنترل کنندهٔ ***PID*** و ***ANFIS*** را وقتی گشتاور با ***N-m*** ۰.۱ است نشان می دهد. جدول *4.3* نیز عملکرد هر دو کنترلر را در گشتاور ***N-m*** ۰.۱ بررسی کرده است.

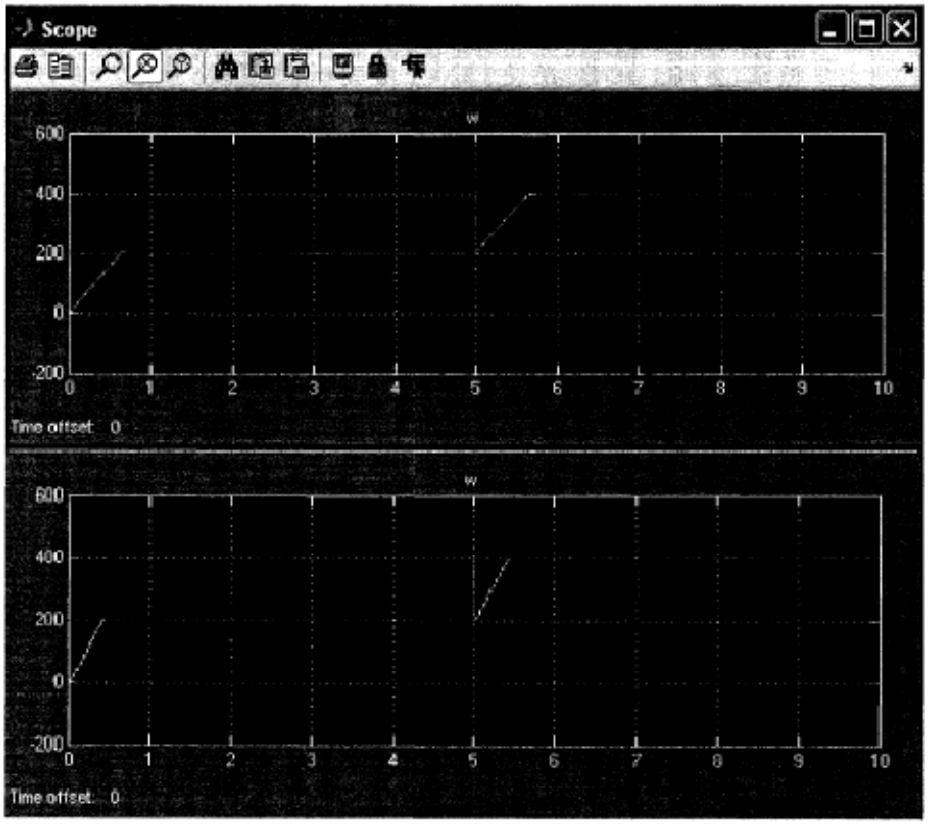

Figure 4.4. Speed variations with PID and ANFIS controllers when torque load is 0.1

Factor	PID	ANFIS
Rise time	0.65 sec	0.45 sec
Overshoot @	203	202.5
Settling time	1 sec	0.45 sec
Steady state error	0	~1 rad/sec

Table 4.3. Analysis of PID and ANFIS controllers at T_L=0.1N-m.

وقتی گشتاور بار ***N-m*** ۰.۲باشد ؛

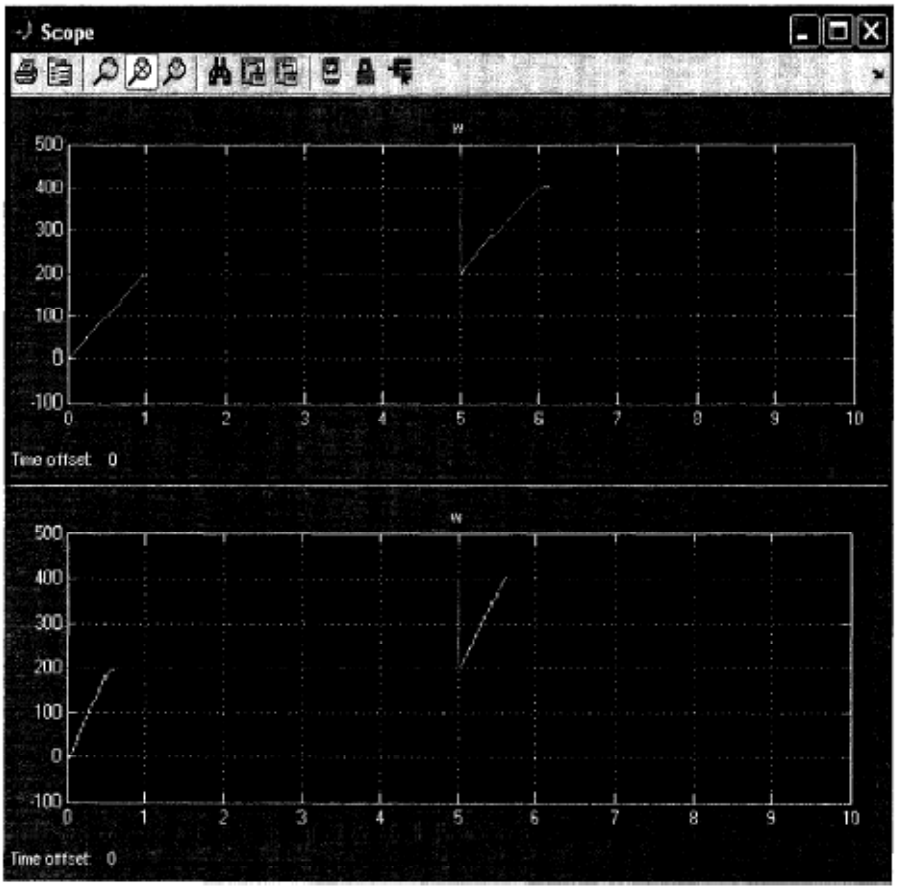

Figure 4.5. Speed variations with PID and ANFIS controllers when torque load is 0.2

Factor	PID	ANFIS
Rise time	1 sec	0.5 sec
Overshoot/ Undershoot	Overshoot at 201	Undershoot at 199
Settling time	1.5 sec	0.65 sec
Steady state error	0	~ 0.5 rad/sec

Table 4.4. Analysis of PID and ANFIS controllers at T_L=0.2N-m.

شکل *4.5* نتایج شبیه سازی استفاده از کنترل کنندهٔ ***PID*** و ***ANFIS*** را وقتی گشتاور با ***N-m*** ۰/۲ است نشان می دهد. همچنین جدول *4.4* عملکردهر دو کنترل کننده ***PID*** و ***ANFIS*** را بررسی کرده است.

وقتی گشتاور بار ***N-m*** ۰.۳ باشد؛

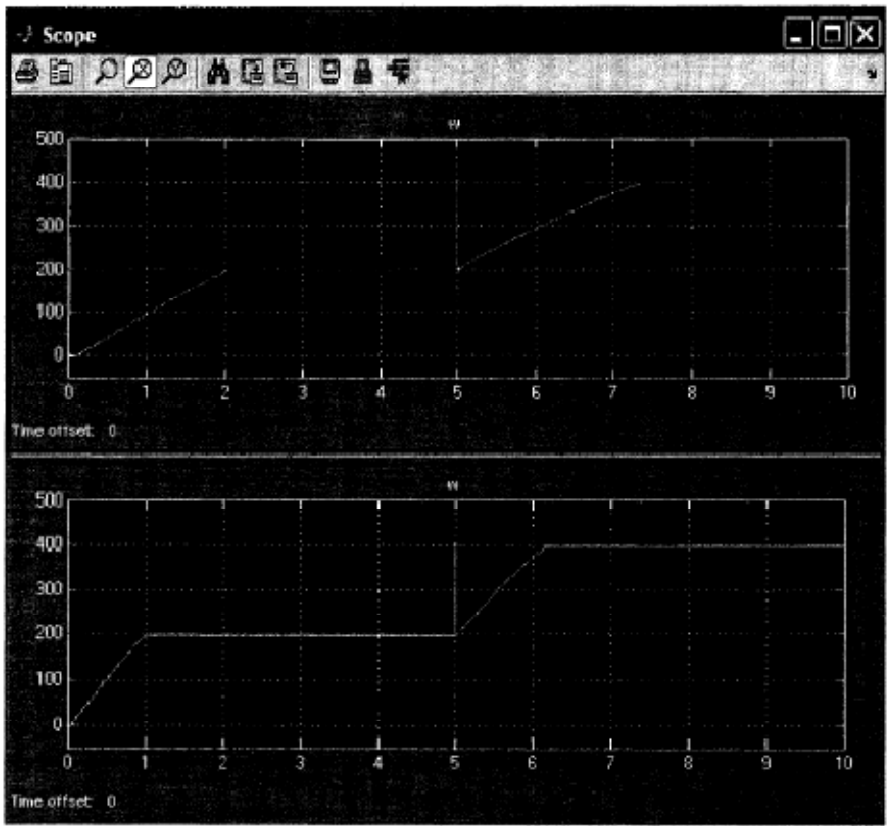

Figure 4.6. Speed variations with PID and ANFIS controllers when torque load is 0.3(

Factor	PID	ANFIS
Rise time	2 sec	1 sec
Overshoot/ Undershoot	-	Undershoot at 197
Settling time	2 sec	1 sec
Steady state error	0	~ 2 rad/sec

Table 4.5. Analysis of PID and ANFIS controllers at T_L=0.3N-m.

نتایج شبیه سازی استفاده از کنترل کننده ***PID*** و ***ANFIS*** وقتی گشتاور بار ***N-m*** ۰.۳ باشد. درشکل *4.6* نشان داده شده است. همچنین جدول *4.5* عملکرد هردو کنترل کننده را وقتی گشتاور بار ***N-m*** ۰.۳ اعمال شده بررسی کرده است.

عملکرد کنترل کنندهٔ ***PID*** و ***ANFIS*** در گشتاور بارهای مختلف با استفاده ازجدول های *4.3* و *4.5* قابل ارزیابی است.

اگر از لحاظ خطای حالت ماندگار ***(Stady state error)*** دو کنترلر را بررسی کنیم. ***PID*** عملکرد بهتری نسبت به ***ANFIS*** دارد .خطای حالت ماندگار برای ***PID*** درگشتاورهای بار مختلف برابر صفر است در

حالی که این برای ***ANFIS*** یک عدد کوچک است .اگر از لحاظ اورشوت و آندرشوت دو کنترلر را بررسی کنیم ملاحظه می شود تفاوت چندانی در رفتار کنترلر ***ANFIS*** و ***PID*** وجـود نـدارد. ازنظـر فاکتورهـای دیگر کنترلی نظیر زمـان خیـز (***rise time***) و زمـان نـزول (***settling time***) ، ***ANFIS*** عملکـرد بهتـری نسبت به ***PID*** دارد .

۴-۳. عملکرد کنترلرهای *ANFIS* و *PID* در گشتاورهای بار آشفته

در این قسمت عملکرد دو کنترلـر ***ANFIS*** و ***PID*** در گشـتاور بارآشـفته یعنـی گشـتاور بـار بـا تغیـرات ناگهانی (پارازیتی) مورد بررسی قرار می گیرد.

برای این کار دو گشتاور بار ،یکـی ***N-m*** ۰.۴ در ***sec*** ۲= ***t*** و دیگـری ***N-m*** ۰.۳ در ***sec*** ۷.۵= ***t*** را در نظـر می گیریم.شکل *4.7* مدل سمبولیک سیستم قبل از اعمال گشتاور آشفته را نشان می دهد که سیسـتم تحت گشتاور بار ثابتی می باشد.

برای اینکه گشتاور بار با تغییـر ناگهـانی را مـدل سـازی کنـیم از بلـوک ***singnal Builder*** در کتابخانـه سمبولیک استفاده می کنیم بطوریکه ***singnal Builder*** دو سیگنال گشتاور بار ***N-m*** ۰.۴ در ***sec*** ۲= ***t*** و گشتاور بار ***N-m*** ۰.۳ در ***sec*** ۷.۵= ***t*** را به گشتاور بار سیستم اضافه می کند. دیاگرام سمبولیک با اضـافه شدن ***singnal Builder*** در شکل *4.7* نشان داده شده است.

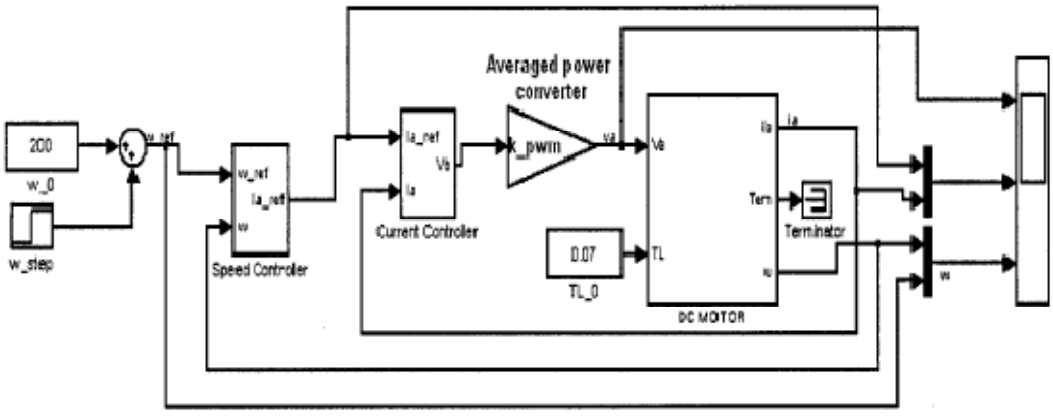

Figure 4.7. Simulink diagram

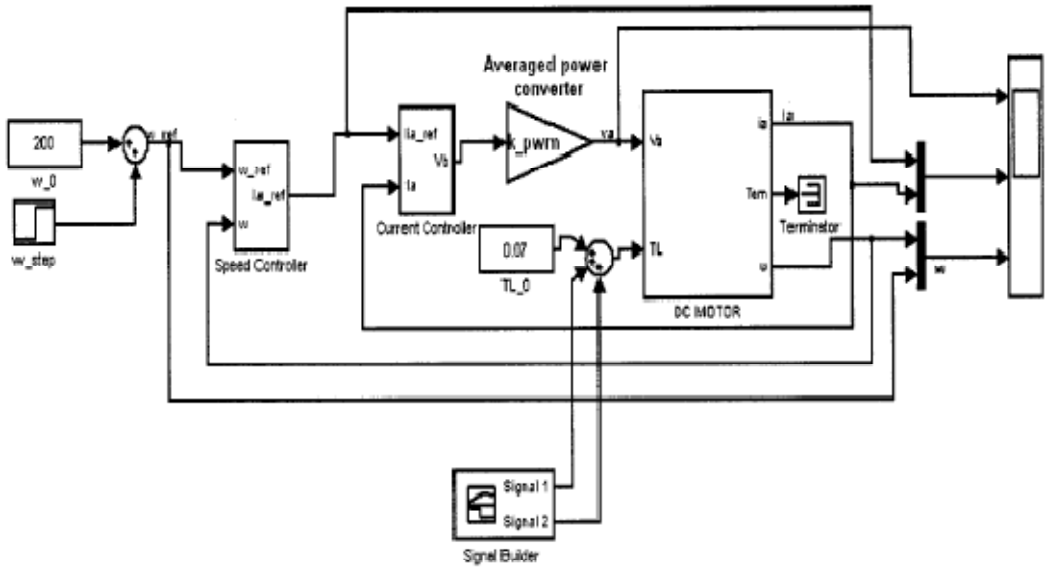

Figure 4.8. Simulink diagram with a signal builder

همانطور که شکل 4.8 نشان می دهد دو سیگنال از ***singnal Builder*** به گشتاور بار اضافه می شوند که در واقع یک ورودی ثانوی به موتور ***DC*** در حال کار می دهد.

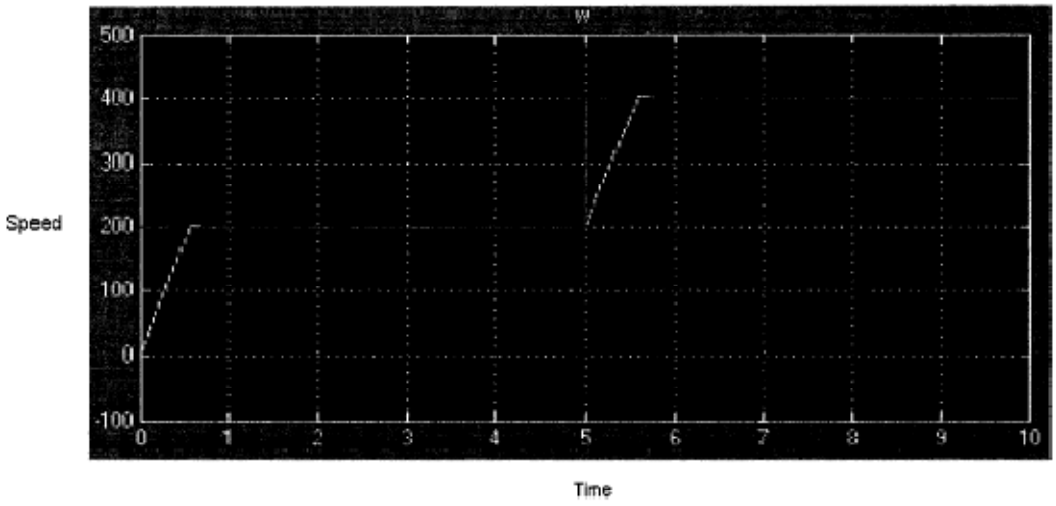

Figure 4.9. Speed response of PID controllers without load disturbances.

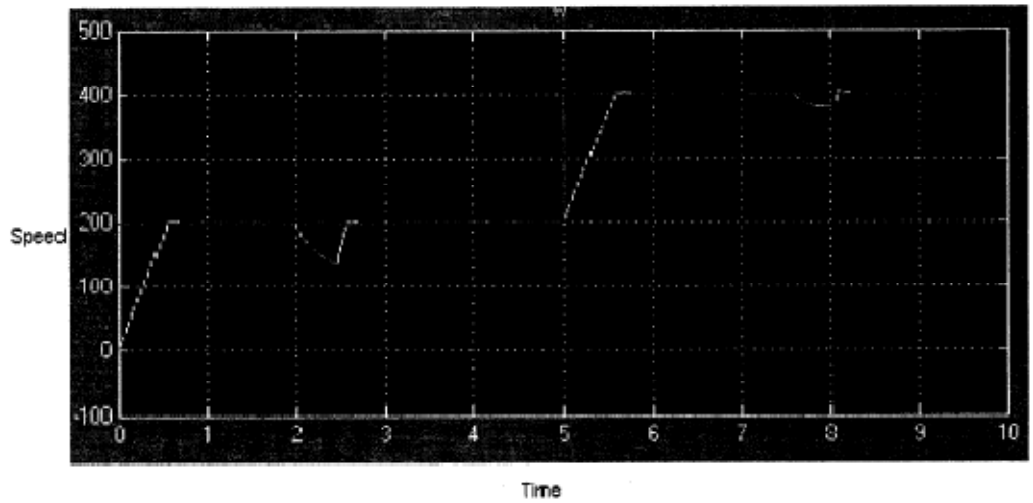

Figure 4.10. Speed response of PID controllers with load disturbances.

تغییرات سرعت درکنترل کنندهٔ ***PID*** بار بدون گشتاور آشفته به ترتیب در شکل های *4.10* و *4.9* نشان داده شده است. در واقع شکل *4.9* تغییرات سرعت بدون ***singnal Builder*** و شکل *4.10* تغییرات سرعت با ***singnal Builder*** را نشان می دهد . درسیستم ***singnal Builder*** در واقع در مرحلهٔ اول گشتاور ***N-m*** ۰.۴ در *2* ثانیه به گشتاور بار اضافه می شود و در مرحلهٔ دوم گشتاور ***N-m*** ۰.۳ در *7.5* ثانیه به گشتاور بار اضافه می شود.

تغییرات سرعت در کنترل کننده ***ANFIS*** بار بدون گشتاور آشفته به ترتیب در شکل های *4.12* و *4.11* نشان داده شده است.

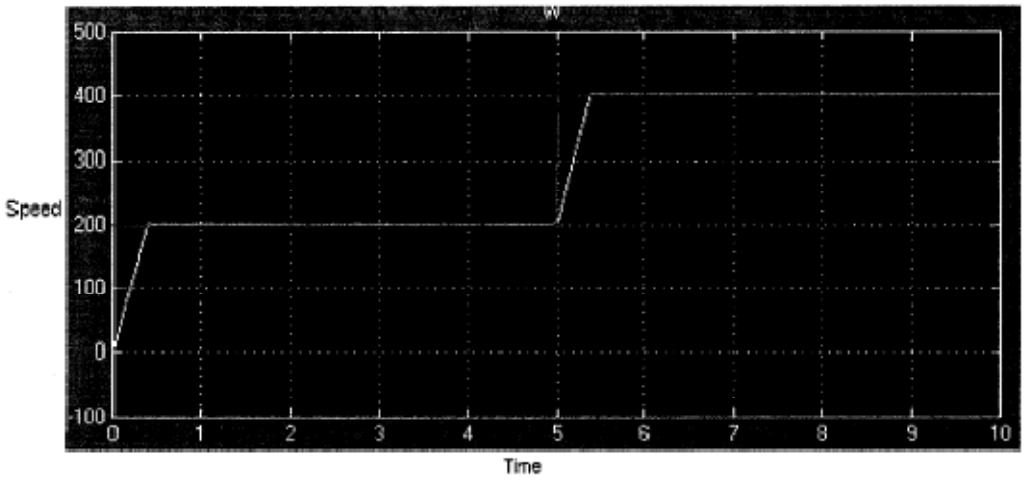

Figure 4.11. Speed response of ANFIS controllers without load disturbances

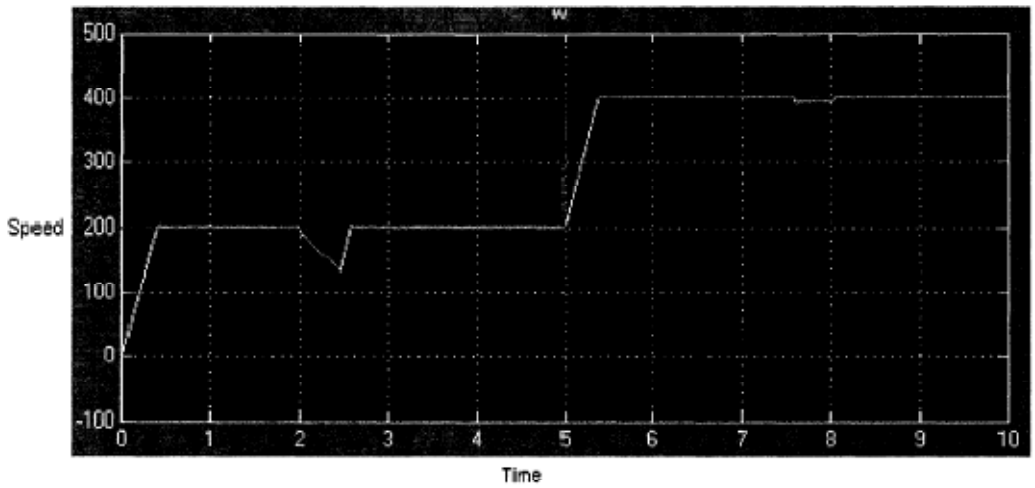

Figure 4.12. Speed response of ANFIS controllers with load disturbances

با مشاهده شکل های *4.10* و *4.12* ملاحظه می شود که برای گشـتاور بـار ***N-m*** ۰.۴ رفتـار دو کنتـرل کننده ***PID*** و***anfis*** شبیه به هم می باشند ولی برای گشتاوربار ***N-m*** ۰.۳ ، ***ANFIS*** عملکرد بهتری دارد.

۵. اجرای سیستم در زمان واقعی (*real- time*)

مدل سیمولینک کنترل کنندهٔ سرعت و جریان از نوع ***ANFIS*** در زمان واقعی در شکل زیـر نشـان داده شده است

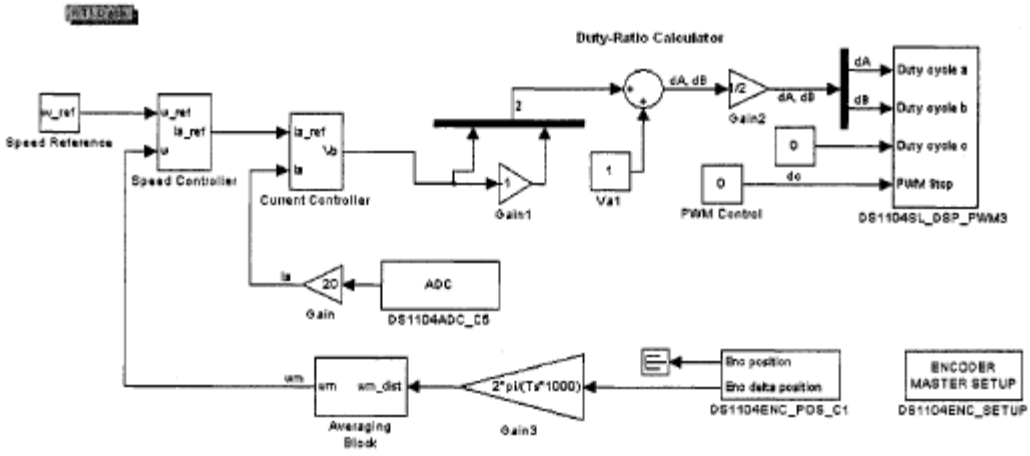

Figure 5.1. Real-time simulink model

کنترل کننده های سرعت و جریان موجود در این شکل همانند چیزی است که در فصـل هـای قبـل در مورد آنها بحث شد.

MATLAB کتابخانه ای به نام ***dsPACE RTI 1104*** دارد که این کتابخانه شامل مجموعه ای از بلوکهای شبیه سازی می باشد. این بلوکها برای ساختن مدل شبیه سازی در زمان واقعی به کار می رود. بلوکهای ***Duty ratio calculator*** و ***Pulse width modulator*** ***(DS1104 – DSP – PWM3)*** جایگزین موتور***DC*** در مدل سیمولینک می شوند .بلوک ***Duty ratio calculato*** کمک می کند تا از ولتاژ متوسط بگیریم قبل از اینکه ولتاژ را مستقیماً به موتور بدهیم.

فرکانس کـاری ***(DS1104 – DSP – PWM3)*** روی ***KHz*** ۵۰ و ***dead band*** آن روی صـفر تنظـیم شـده است. این بلوک مسئول فرستادن پالس به ***Lab _ Kit DC motor*** می باشد.

Duty ratio در سیمولینک از فرمول های زیر ساخته می شود:

$$V_{control,A} = -V_{control,B} = Vmotor/V_d \text{ , where } V_d \text{ is Bus voltage} = 1.$$

$$d_A=1/2(V_{control,A} + 1)$$

$$d_B=1/2(V_{control,B} + 1).$$

دیاگرام ***Duty ratio calculato*** در سیمولینک که موازی با ***PWM*** قرار دارد در شکل زیر نشان داده شده است.

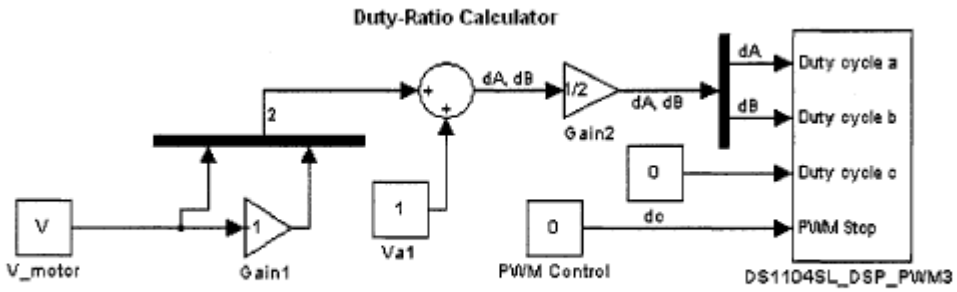

Figure 5.2. Simulink diagram of duty ratio calculator along with

بلوک ***Encoder Master setup (DS 1104 ENC – SETUP)*** برای محاسبهٔ سرعت و موقعیت موتور استفاده میشود . سرعت با استفاده از بلوک ***DS 1104 ENC- POS – C1*** بر حسب ***RPM*** اندازه گیری م شود . اما سرعت باید بر حسب ***Rad/ sec*** باشدتا توسط ***dsPACE*** قابل خواندن به طور صحیح باشد . بنابراین بایدخروجی ***DS 1104 ENC- POS – C1*** را به یک بلوک مثلثی شکل بهره ***(Gain)*** وصل کرد که بهره آن برابر $\pi ۲/۱۰۰۰ Ts$ می باشد و در آن ***Ts*** زمان نمونه برداری است.

بلوک ***averaging block*** با تأخیرهایی که ایجاد می کند از نویزی شدن سیگنال سرعت جلوگیری می کند . در واقع اگر مقدار سرعتی که به این بلوک داده می شود در اثر نویز تغییر کند این بلوک با تأخیری که ایجاد می کند متوسط موج را می گیرد(**موج** $\frac{1}{T}\int_{T}^{\text{ورودي}}$ که در آن T زمان تاخیر است) و به خروجی می برد. یک ***averaging block*** دوازده تاخیری طراحی شده و به عنوان زیر سیستم به مدل اضافه شده است. *(به شکل 5.1 نگاه شود .)*

زمان نمونه برداری واحد تاخیری روی *1-* تنظیم شده است . ***averaging block*** کمک می کند تا نویز را کاهش دهیم . استفاده از این بلوک دربسیاری از کاربردهای زمان واقعی بسیار معمول است . دلیل کم شدن نویز به واسطه ***averaging block*** این است که مقدار متوسط نویز همواره صفر است. بنابراین اگر موج نویزی شده را به صورت موج اصلی بدون نویز به اضافه موج نویز در نظر بگیریم با عمل متوسط گیری نویز حذف می شود.

averaging block طراحی شده در سیمولینک در شکل زیر نشان داده شده است:

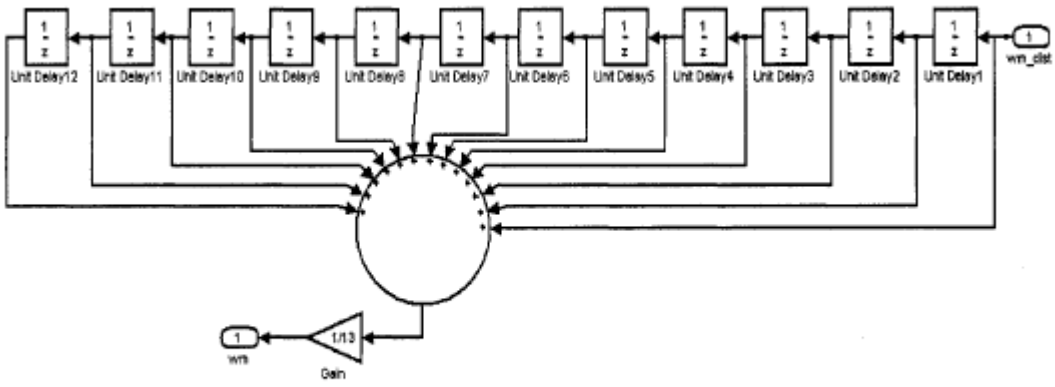

Figure 5.3. Averaging block model in simulink

بلوک ***DS 1104 ADC-C5***برای تبدیل آنالوگ به دیجیتال استفاده مـی شـود .ایـن بلـوک بـرای گـرفتن مقدار جریان واقعی از موتور استفاده می شود. مقدار جریانی که گرفته می شود بـه وسـیلۀ بلـوک ***Gain*** قبل از دادن آن به کنترل کننده جریان تغییر اندازه داده می شود. نسبت تبدیل *1:10* می باشد.

یک ورودی شیب به عنوان سرعت مرجع به سیستم داده می شود که در آن شـیب از صـفر تـا ***red/sec*** ۲۰۰ در ***t***=۱۰***sec*** و از *200* تا ***red/sec*** ۴۰۰ در ***sec*** ***t***=۱۵ تغییر می کند .زمـان بـین صـفر تـا بینهایـت تنظیم می شود.

سرعت مرجع با یک سیگنال شیب- ثابت در عوض شیب – متغیر به سیستم داده مـی شـود .یکبـار کـه مدل ساخته شد، برای اجرا آماده می شود.

شبیه سازی مدل سیمولینک در زمان واقعی چیزی شبیه شـکل *5.4* اسـت .زمـان نمونـه بـرداری روی ***0.0001sec*** تنظیم شده است.

مراحل ساختن مدل زمان واقعی به صورت زیر است :

Tools → Real – Time work shop → Build model

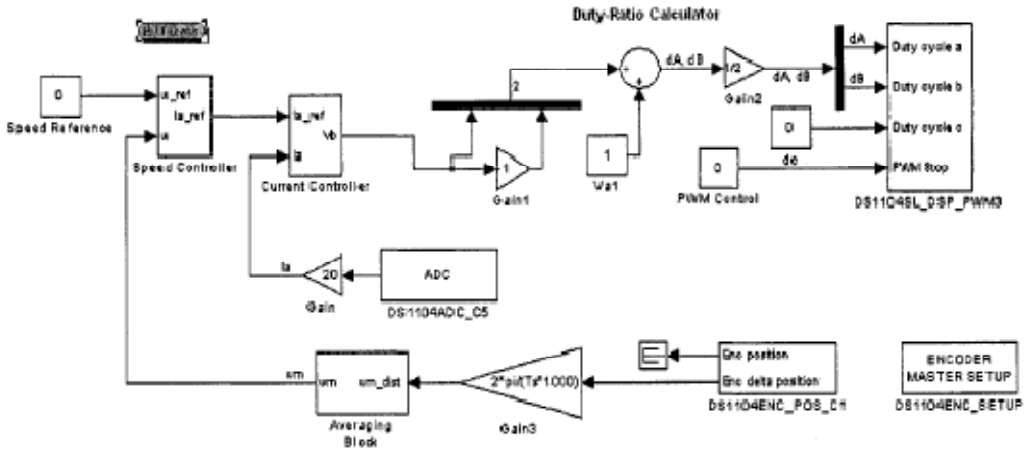

Figure 5.4. Real-time simulink model

وقتی که کار ساخته شدن مدل در زمان واقعی تمام شد ، تأییدیهٔ با موفقیت تمام شدن کار در ***MATLAB command*** ظاهر می شود . شکل زیر تأییدیهٔ ساخته شدن زمان واقعی را نشان می دهد.

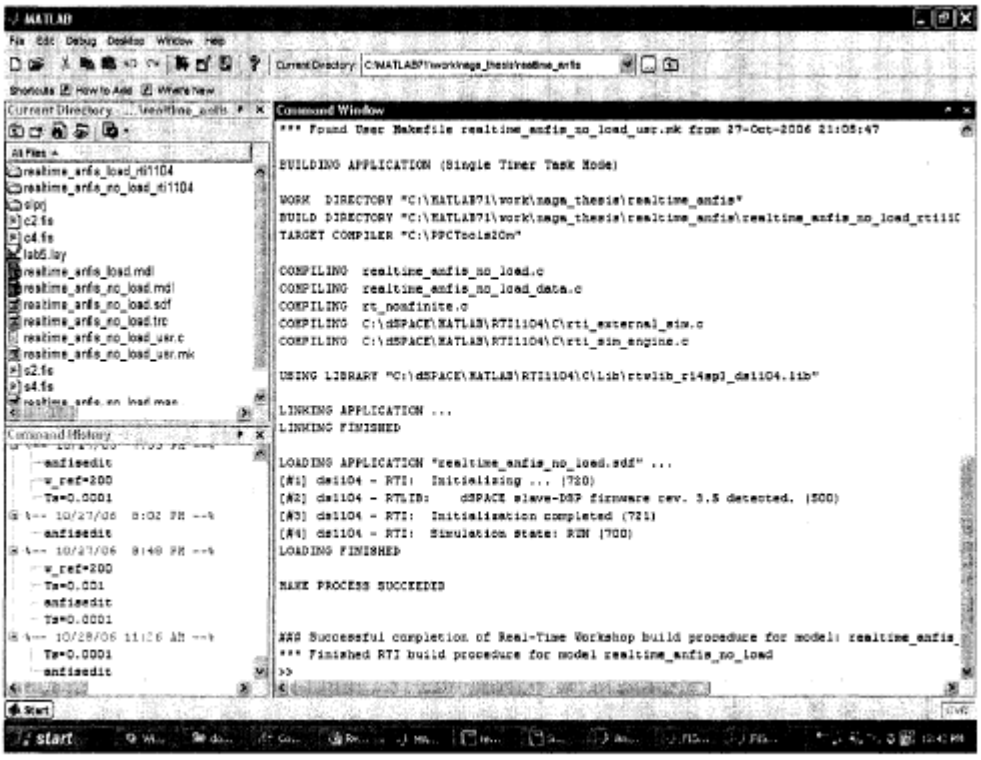

Figure 5.5. Successful completion of building real-time workshop.

وقتی کارساخته شدن مدل در زمان واقعی به پایان رسید این به این معنی است که همهٔ فایل های مورد نیاز برای اجرا در حافظهٔ ***DSP*** بار شده و اجرا شروع می شود. ما از ***dsPACE control desk*** به منظور مشاهده کردن ، اصلاح کردن و آنالیز متغیرها استفاده می کنیم . ***dsPACE control desk*** یک ***GUT*** می باشد که شبیه شکل *5.6* است .

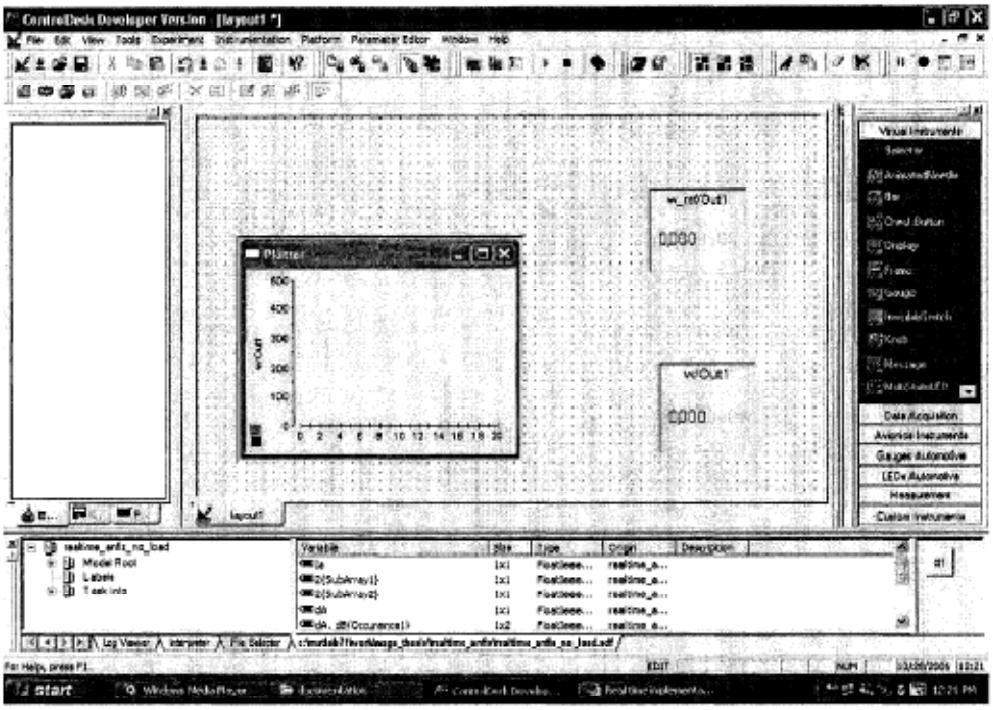

Figure 5.6. Control desk layout

۵-۱-۱. نتایج اجرا در زمان واقعی کنترلر *ANFIS*

وقتی تنظیمات در پنجره ***control desk*** انجام شد، با کلیک کردن روی دکمهٔ ***play*** اجـرای زمـان واقعـی شروع می شود . نتایج اجرا زمان واقعی کنترل کننـده ***ANFIS*** در شـکل *7.5* نشـان داده شـده اسـت . همانطور که دیده می شود پاسخ زمانی بسیار خوب و خطای حالت ماندگار حدود ***red/ sec*** ۴ می باشد.

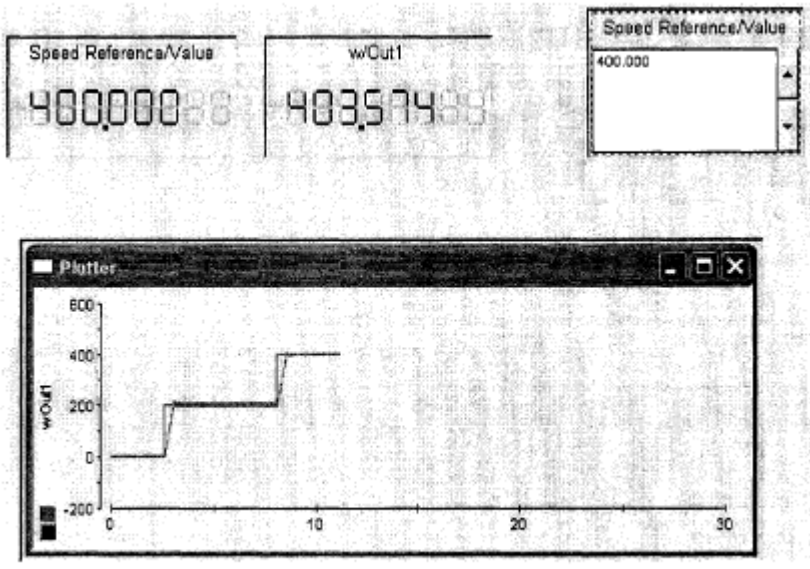

Figure 5.7. Results of using ANFIS controllers

نتایج استفاده از کنترل کنندهٔ ***ANFIS*** برای سرعت های مختلف در پیوست ***D*** آمده است.

۵-۲. مدل سیمولینک در زمان واقعی کنترل کنندهٔ *PID*

مدل سیمولینک در زمان واقعی کنترل کننده ***PID*** شبیه مدل زمان واقعی ***ANFIS*** می باشد با این تفاوت که کنترل کننده های سرعت و جریان موجود در مدل به جای اینکه از نوع ***ANFIS*** باشند ، از نوع ***PID*** هستند.برای کنترل کننده های جریان وسرعت یک ورودی ***reset*** اضافه شده است. زمان نمونه برداری روی 0.001 و زمان شبیه سازی از صفر تا بینهایت تنظیم شده است .سرعت مرجع به صورت یک سیگنال شیب ثابت می باشد. مدل سیمولینک کنترل کنندهٔ ***PID*** در زمان واقعی در شکل 8.5 نشان داده شده است.

۵-۲-۱. نتایج تحلیل زمان واقعی کنترل کنندهٔ *PID*

نتایج شبیه سازی زمان واقعی استفاده از کنترل کنندهٔ ***PID*** در شکل 5.9 نشان داده شده است. همانطور که از روی شکل ملاحظه می شود می توان دریافت که پاسخ زمانی کنترل کنندهٔ ***PID*** کمی کندتر از کنترل کنندهٔ ***ANFIS*** می باشد. این در حالی است که خطای حالت ماندگار ***PID*** کمتر از ***ANFIS*** و حدود ***red/sec*** ۱ می باشد.

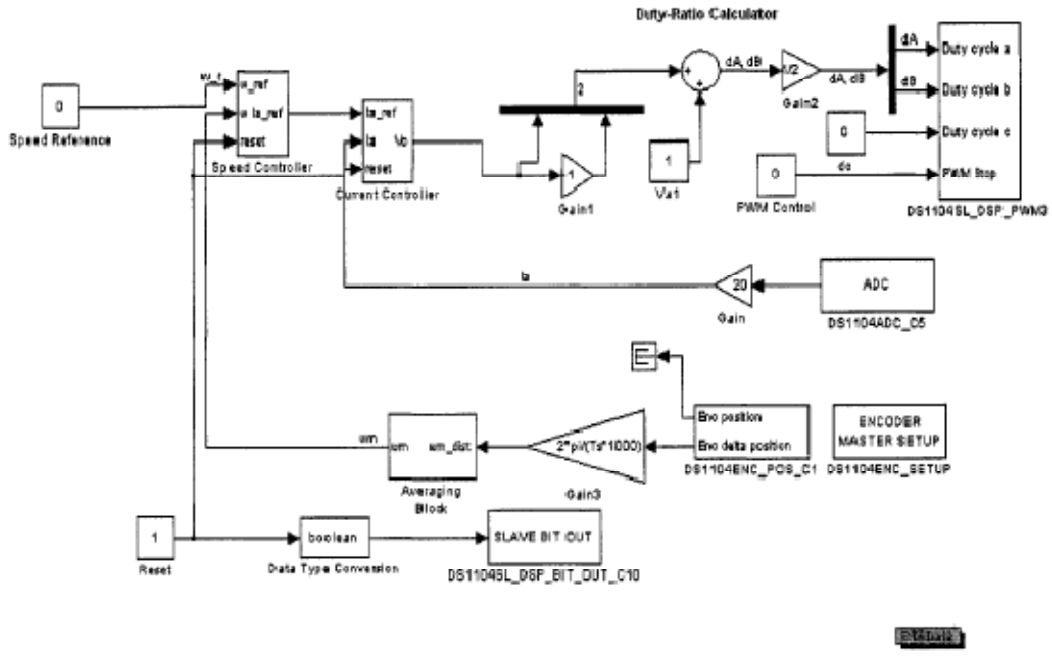

Figure 5.8. Real-time simulink model using PID speed and current controllers.

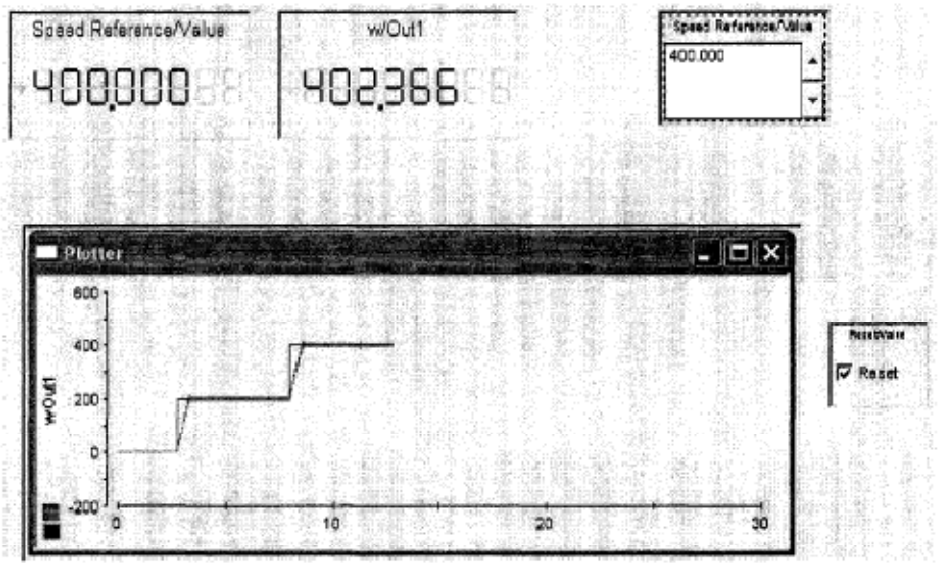

Figure 5.9. Real-time results of using PID controllers.

نتایج شبیه سازی کنترل کننده ***PID*** در زمان واقعی برای سرعت های مختلف در پیوست ***D*** آمده است.

۵-۳. نتایج شبیه سازی زمان واقعی برای بارهای آشفته (پارازیتی)

در این قسمت یک بار آشفته به موتور اعمال می شود و نتایج کنترل کننده های ***PID*** و ***ANFIS*** مشاهده می گردد.شکل *5.10* و *5.11* به ترتیب پاسخ زمانی کنترل کنندهٔ ***ANFIS*** و ***PID*** را برای یک بار آشفته نشان می دهد. این نتایج بدست آمده در حالی است که موتور از قبل با سرعت ***red/sec*** ۵۵ در حال حرکت بوده است. محور ***x***ها معرف زمان و محور ***y*** ها معرف سرعت می باشد.

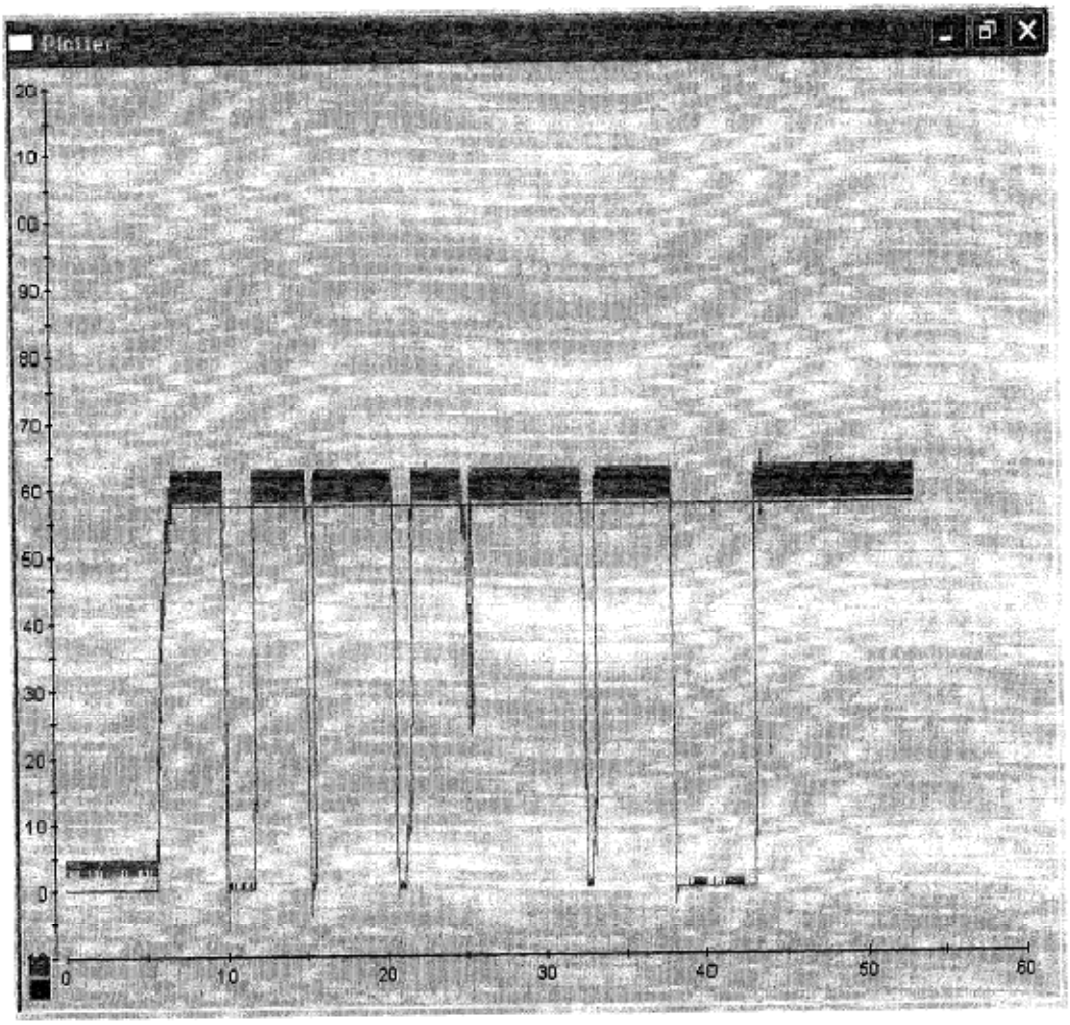

Figure 5.10. Real-time results of using ANFIS controllers with load disturbances

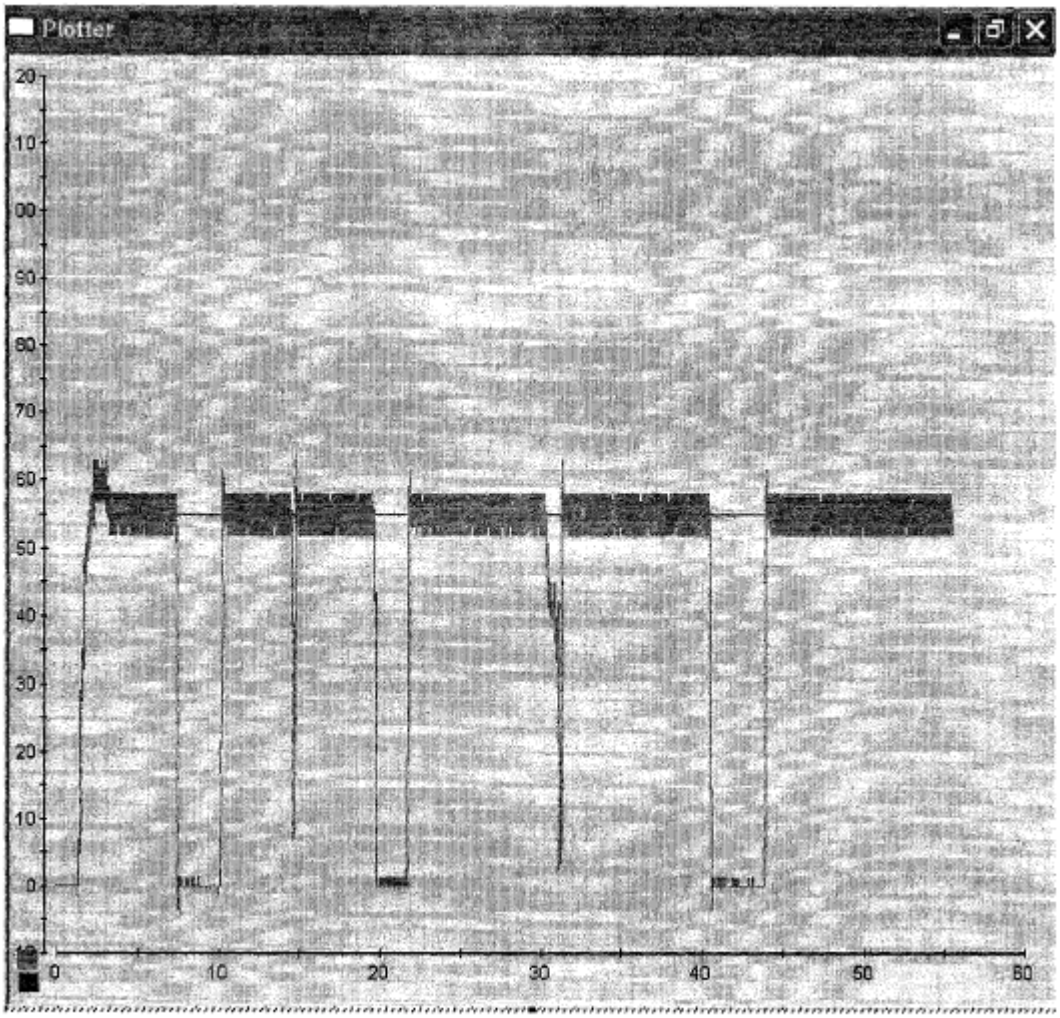

Figure 5.11. Real-time results of using PID controllers with load disturbances

همانطور که از نتایج پاسخ زمانی کنترل کننده ***PID*** مشاهده می شود این کنترل کننده وقتی بار آشفته اعمال می شود دارای اورشورت هایی در شکل موج می باشد. نتایج پاسخ زمانی کنترل کننده ***ANFIS*** برای بارهای آشفته نشان می دهد که شکل موج با وجود اینکه یک خطای حالت ماندگار کوچک وجود دار ولی اورشوتی وجود ندارد .بنابراین در جاهایی که بارهای آشفته وجود دارد عملکرد کنترلر ***ANFIS*** خیلی بهتر از ***PID*** می باشد.

فصل ششم

۶. نتیجه گیری

نتایج این پایان نامه در کنترل موتور ***DC*** با استفاده از سیستم استنتاج فازی عصبی تطبیقی (***Adapive Neuro- Fuzy INFERENCE Sysem***) مفید و موفقیت آمیز بوده و کنترل کنندۀ ***ANFIS*** با موفقیت با یادگیری از کنترل کننده ***PID*** طراحی شد.

شبیه سازی ها در ***MATLAB/ Simulink*** انجام شده و عملکرد سیستم در زمان واقعی با استفاده از ***dsPACE*** دانلود شده و به وسیلۀ ***dsPACE*** اجرای زمان واقعی آن انجام شد.

عملکرد هر دو کنترلر ***ANFIS*** و ***PID*** درگشتاورهای بار مختلف ارزیابی شد و ملاحظه گردید که از لحاظ فاکتورهای زمان خیز و زمان نزول سیستم ، ***ANFIS*** عملکرد بهتری داشت.

از لحاظ فاکتور خطا حالت ماندگار ***PID*** عملکرد بهتری نسبت به ***ANFIS*** داشت . خطای حالت ماندگار ***ANFIS*** حدود ***red/sec*** ۴ بود. گشتاور آشفته به صورت دو گشتاور ۰/۳***N-m*** و ۰/۴ ***N-m*** مدل سازی شد و نتایج در شبیه سازی ملاحظه گردید و در یافتیم که عملکرد دو کنترلر در گشتاور ۰/۴***N-m*** شبیه به هم بود در حالی که درگشتاور ۰/۳ ***N-m*** عملکرد ***ANFIS*** کمی بهتر از ***PID*** بود.

همچنین عملکرد ***PID*** و ***dsPACE*** مورد ارزیابی قرار گرفت و نتیجه گرفتیم که از نظر خطاهای حالت ماندگار ***PID*** در حد ناچیز از***ANFIS*** چشم پوشی کنیم. در کل عملکرد در زمان واقعی ***ANFIS*** بهتر از ***PID*** می باشد .اما باید توجه داشت اگر روی خطای حالت ماندگار خیلی حساس باشیم بهتر است از ***PID*** استفاده کنیم.

۶-۱. کارهای بعدی

ANFIS برای کنترل سرعت موتور ***DC*** همانطور که هدف این پایان نام بود طراحی شد.کارها و تحقیقات بیشتری در مورد ***ANFIS*** در آینده می توان انجام داد که تعدادی از آنها در زیر لیست شده اند:

کنترل کنندۀ ***ANFIS*** که در این پایان نامه بحث شد فقط دارای یک ورودی یعنی خطای بین مقدار مرجع و مقدار واقعی بود. کنترل کنندۀ ***ANFIS***میتواند به دو ورودی یکی خطا و دیگر مشتق گیر توسعه یابد که در این صورت خطای حالت ماندگار مینیمم می شود.

کنترل کنندهٔ ***ANFIS*** می تواند برای کنترل سرعت موتور ***DC*** درنوسانات بار ناشناخته و رندوم نیز طراحی شود.

الگوریتم یادگیری هیبریدی که ازترکیب الگوریتم های حداقل مربعات و تندترین شیب ایجادشده بود در این پایان نامه استفاده شد. اما قواعدهای یادگیری پیشرفته تری وجوددارد که می توان با ترکیب آنها با منطق فازی عملکرد بهتری را برای سیستم به ارمغان آورد.

کنترل کنندهٔ ***ANFIS*** را می توان برای الکتروموتورهای مختلف نظیر موتور ***ac*** ، استپرموتور ، موتورهای القایی نیز به کار برد.

پیوست A

پارامترهای موتور DC

پارامترهای موتور ***DC*** که در یک ***m-file*** ذخیره می شود قبل از شبیه سازی اجرا می شود . در شکل زیر ***m-file*** مربوطه به یک موتور ***DC*** رانشان میدهد.

Figure A1. DC motor parameters

پارامترهای موتور ***DC*** با انجام آزمایش های روی سیستم کنترل حلقه باز موتور ***DC*** بار و بدون بار بدست می آید.

پارامترهای حالت ماندگار ازروی مشخصهٔ گشتاور سرعت موتور ***DC*** همراه با بار محاسبه می شوند. اندوکتانس و اینرسی از آنالیز پاسخ گذاری ماشین تعیین می شوند.

پیوست B

مقدمه ای بر dsPACE

dSPACE یک بک کنترلی بسیار قوی می باشد کـه بـا پردازنـدهٔ ***TMS3220C31 DSP*** سـاخت شـرکت تگزاس کار می کند. سیمولینک میتواند مستقیماً با ***dSPACE*** ارتباط برقرار کند . ***dSPACE*** این امکـان را فراهم میکند تا طرح مدل شده در سیمولینک را مانند یک نمونه اولیه ساخته شـده از طـرح آزمـایش کنیم. این کار با دانلو کردن مدل سیمولینک در ***DSP*** انجام می شود.

ویژگی های اصلی ***dSPACE*** عبارتند از :

۱- ***dsPACE*** یک سیستم بامعماری باز (***Open architectar***) می باشد .بدین معنی که هر ماشین با هـر الگوریتم کنترلی را می توان با آن مورد آزمایش قرار داد.

۲- پیکربندی نرم افزاری ***dsPACE*** : بدین معنی که با اصلاح کردن ساده مدل سیمولینک ، هر موتـور را با هر حلقهٔ کنترلی میتواند درایو کند.

۳- نرم افزار ***dsPACE*** نیازی به دانستن زبان های برنامه نویسی نظیر ***C*** و اسمبلی و ..ندارد.

DsPACE شامل سخت افزاری به نام بورد کنترل کننده (***DS1104R8D***) و یک نرم افزار به نام ***control dsPACE desk*** می باشد.

برد کنترلی DS1104R8D:

یک برداستاندارد ***PCI*** است که در اسلات ***PCI*** کامپیوتر جا می خـورد .ایـن بـورد اساسـاً بـرای سـاخت کنترل کننده های دیجیتال با سرعت بالا و شبیه سازی زمان واقعی ساخته شده است. پردازنـده ای کـه در آن به کار رفته از نوع ***PC603e*** می باشد. مشخصات این پردازنده در زیر لیست شده است.

1. High performance super-scalar microprocessor.
2. Five independent execution units and two register files.
3. High instruction data through put.
4. Facilities for enhanced system performance.
5. Integrated power management [17].

همچنین این بورد شامل یک زیر سیستم ***DSP _ slave*** مبتنی بر ***TMS320F240*** برای ارتباط ***I/O*** می باشد. مشخصات ***TMS320F240*** در زیر لیست شده اند:

1. Is the integration of DSP core and micro controller.
2. Can execute 20 million instructions/second.
3. Contains all the peripherals needed to provide single chip stand alone DSP controller.
4. Is designed specially for digital motor control.
5. Controls the motor without the use of any position sensor to give rotor's position and speed [16].

مشخصات بورد کنترلی ***DS1104 R&D*** به شرح زیر است :

1. Master PPC represents the computing power of board.
2. Slave DSP features I/O units in addition to those in Master PPC.
3. Memory comprised of DRAM and flash memory.
4. Timers provide a sample rate timer, a time base counter and 4 general-purpose timers.

شکل زیر معماری و بلوکهای توابع ***DS1104*** را نشان می دهد.

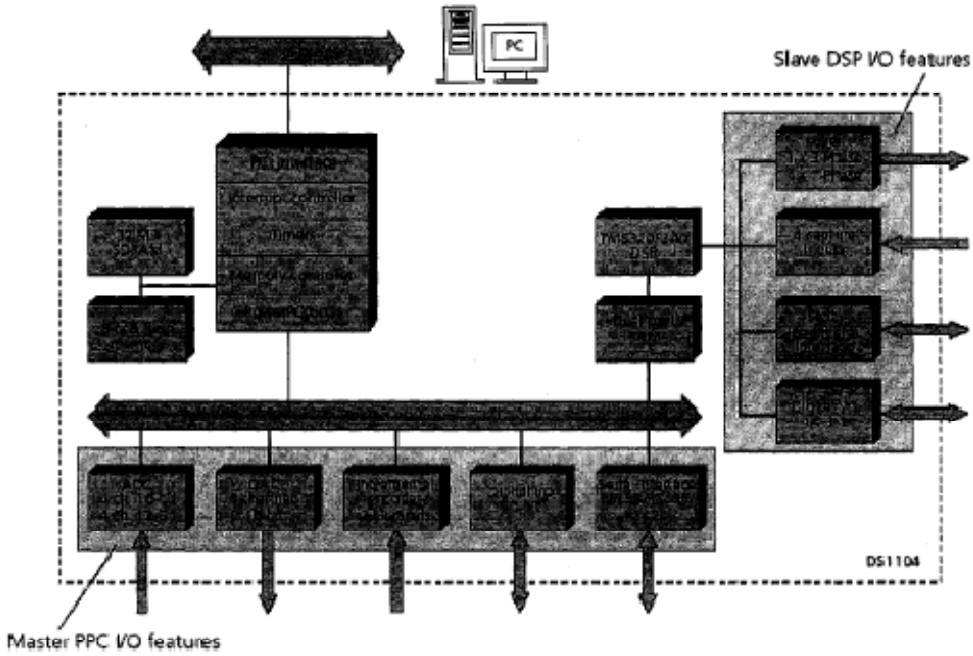

Figure B1. Architecture and functional units of DS 1104 [18].

جدول زیر مشخصات بوردکنترلی ***R&D*** را نشان می دهد.

Parameters	Characteristics
Main Processor	■ MPC8240, PowerPC 603e core, 250MHz ■ 32 kB internal cache
Timers	■ 1 sample rate timer, 32-bit downcounter ■ 4 general purpose timers, 32 bit ■ 64-bit time base for time measurement
Memory	■ 32 MB synchronous DRAM (SDRAM) ■ 8 MB boot fl ash for applications
Interrupt Control Unit	■ Interrupts by timers, serial interface, slave DSP, incremental encoders, ADC, host PC and 4 external inputs ■ PWM synchronous interrupt
Analog Input	■ 4 ADC inputs with one ADC unit, 16 bit, multiplexed

	■ ± 10 V input voltage range ■ 2 µs sampling time *) ■ > 80 dB signal-to-noise ratio ■ 4 ADC channels, 12 bit ■ ± 10 V input voltage range ■ 800 ns sampling time *) ■ > 65 dB signal-to-noise ratio
Analog Output	■ 8 channels, 16 bit, 10 µs max. settling time ■ ± 10 V output voltage range
Incremental Encoder Interface	■ Two digital inputs, TTL or RS422 ■ 24-bit digital incremental encoders ■ Max. 1.65 MHz input frequency, i.e. fourfold pulse counts up to 6.6 MHz ■ 5 V / 0.5 A sensor supply voltage
Digital I/O	■ 20-bit digital I/O (bit-selectable direction) ■ ± 5 mA output current
Serial Interface	■ Serial UART (RS232, RS485 or RS422)
Slave DSP Subsystem	■ Texas Instruments' DSP TMS320F240 ■ 4 kWord of dual-port RAM ■ Three-phase PWM outputs plus 4 single PWM outputs ■ Frequency measurement (F/D) and generation (D/F), 4 channels each ■ 14 bits of digital I/O (TTL)
Physical Characteristics	■ Power supply 5 V, 2.5 A / -12 V, 0.2 A / 12 V, 0.3 A ■ Operating temperature 0 to 55 °C (32 to 131 °F) ■ Requires one 33 MHz / 32-bit 5-V PCI slot ■ The I/O connector can be linked to two 50-pin, female Sub-D connectors using the adapter cable supplied.

Table B1. Specifications of R&D Controller board [19]

نرم افزار *dSPACE*

نرم افزار ***dSPACE*** از نوع نرم افزارهای رابط گرافیکی با کاربر یا همان ***GUI*** می باشد که محیطی گرافیکی را برای کاربر فراهم میکند. ابتدا قبل از اینکه مدل سیمولینک با ***dSPACE*** ارتباط برقرا کند،

باید آن را به صورت مدل زمان واقعی که قبلاً بحث شد ، در آورد. مدل زمان واقعی به وسایل خـارجی از قبیل سیگنال ژنراتور وصل می شود.

نرم افزار ***control desk*** به ما کمک می کند تا متغیرها را در حین اجرای زمان واقعی بینیم و پارامترهـای شبیه سازی شده را مشاهده و آنها را تغییردهیم.

شکل زیر نمونه ای از پنجرهٔ ***control desk*** را نشان میدهد.

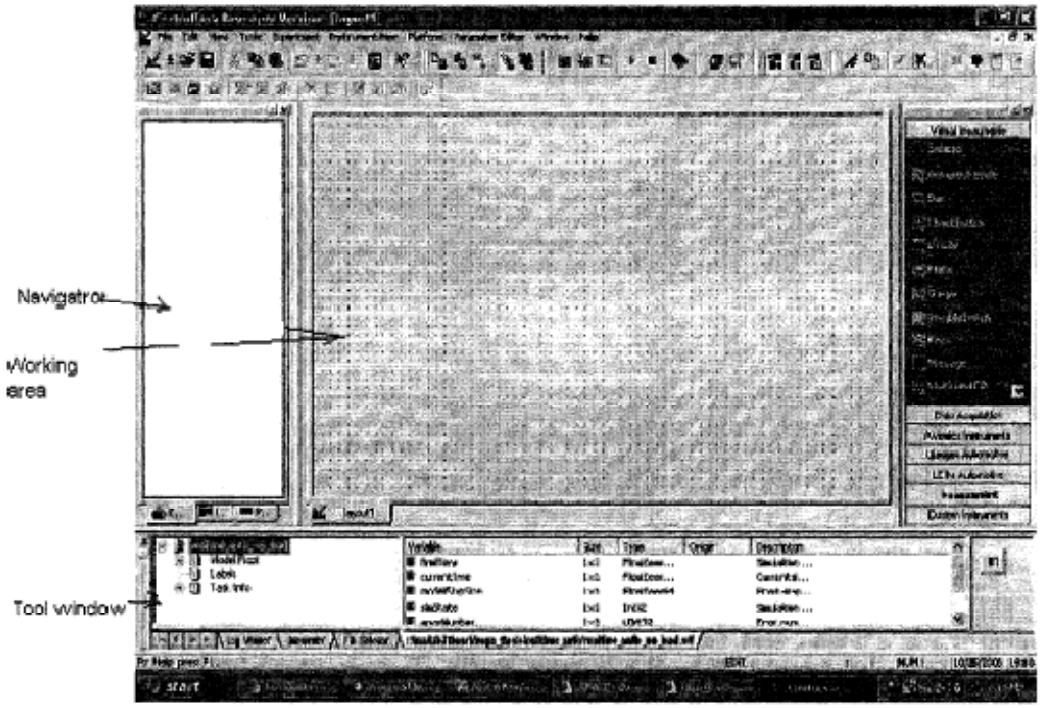

Figure B2. Control Desk window

همانطور که از شکل پیداست ، پنجره نرم افـزار دارای قسـمت ***Navigator*** مـی باشـد کـه سـه محـیط مختلف را برای کاربر می تواند نشان دهد.این سه محیط عبارتنـد از ***expriment viwe ، hardware viwe*** و ***instrument view*** . پنجرهٔ***tools*** برای ابزارهای مختلفی استفاده میشود که در ***Navigator*** انتخاب شده اند.

پنجرهٔ ***tools*** تعدادی ابزار نظیر ***log- view*** و ***Interpreter*** ، ***File selector*** و.. دارد . از بـین ابزارهـا فقـط دو ابزار به نام ***parameter Editor , Variable Browser*** وجود دارند کـه شـامل متغیرهـای اجرایـی مـی باشد. متغیرها با پسوند ***.sdf*** ذخیره می شوند .

سه حالت عبارتند از ***Edit mode , Test mode*** و***Animation mode*** .شکل زیر نوار ابزاری که ساخت فوق رابرای انتخاب کردن در اختیار کاربر می گذارد نشان میدهد.

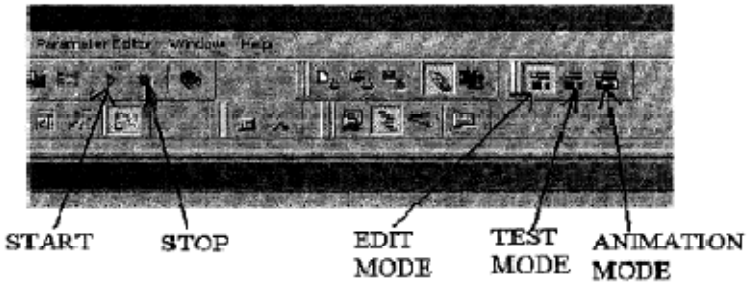

Figure B3. Execution and Animation Control toolbox [18].

دستورالعمل مرحله به مرحله انجام یک آزمایش با ***dSPACE*** در فلورچارت زیرنشان داده شده است. دو مرحله کلی وجود دارد .مرحلۀ اول شامل ساخت یک مدل سیمولینک برای اجـرا در ***dSPACE*** و مرحلـۀ دوم شامل اجرای زمان واقعی می باشد.

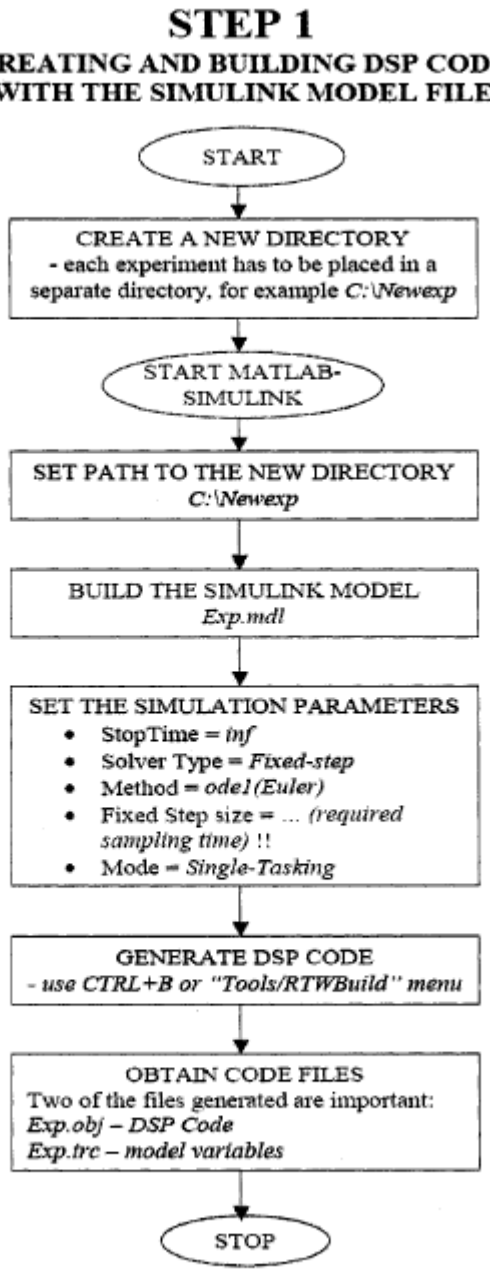

Figure B4. Flow chart for creating and building DSP Code

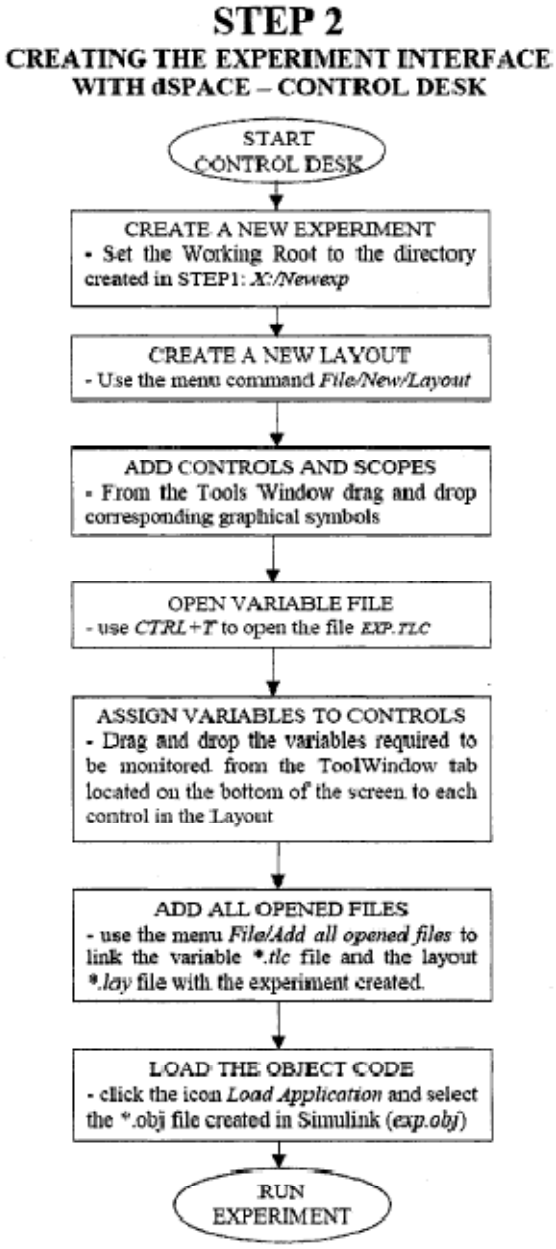

Figure B5. Flow chart for creating an experiment for interface between dSPACE and control desk [18].

(B–۲) : بورد درایو الکتریکی

بورد درایو الکتریکی در بسیاری از آزمایشگاههای ماشین ***ac*** و ***DC*** و آزمایشگاه های درایو الکتریکی استفاده می شود. در زیر ویژگی های بورد درایو لیست شده است.

دو اینورتر ***PWM*** مستقل برای کنترل همزمان چندماشین

ولتاژ باس *42* ولت

کانالهای ورودی ***PWM*** دیجیتال برای طراحی در زمان واقعی

ارتباط کامل دیجیتال و آنالوگ با بورد ***dsPACE***

بلوک دیاگرام یک بورد الکتریکی در شکل زیر نشان داده شده است:

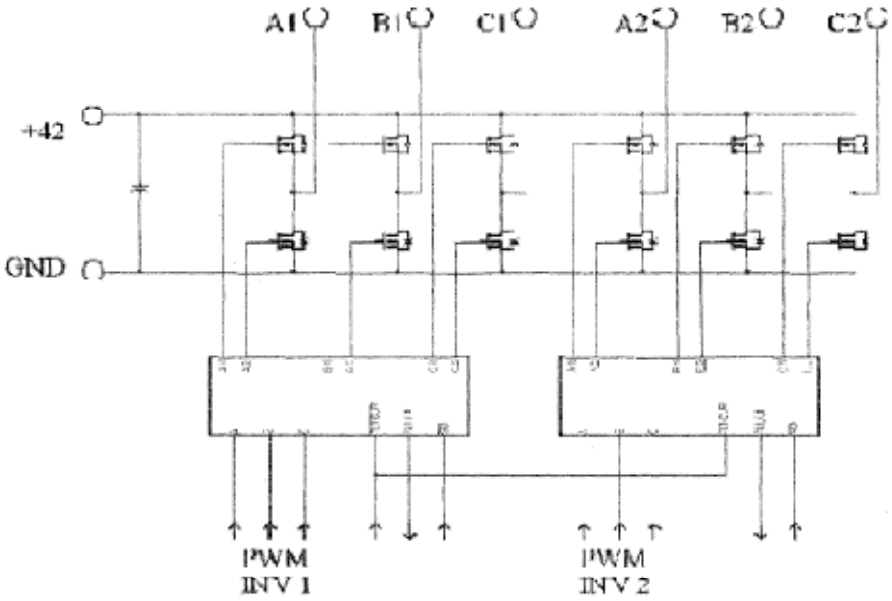

Figure B6. Block diagram of Electric drives board [19].

در دو اینورتر سه فاز استفاده شده دربورد الکتریکی المان سوئیچینگ ***MOSFET*** می باشد. خروجی اینورتر اول با ***A1*** و ***B1*** و ***C1*** علامتگذاری شده و خروجی اینورتر دوم با ***A2*** و ***B2*** و ***C2*** .

جریان خروجی اینورترها با سنسور ***LEM*** اندازه گیری می شود . فقط جریان های فازهای ***A*** و***b*** اندازه گیری می شود و جریان فاز ***C*** از فرمول زیر بدست می آید:

$$Ia + Ib + Ic = 0$$

سنسور به ازای عبور جریان *1* آمپر *0/5* ولت در خروجی خود ظاهر می کند.

* حفاظت در برابر فالت : بوردهای درایو شامل حفاظت اضافه جریان برای هر اینورتر می باشند. ***LED*** قرمز رنگ نشان دهندهٔ اضافه جریان در اینورتر می باشد . با فشردن دکمه ***reset*** می توان ***LED*** را ***reset*** کرد.

TRAINING DATA FOR ANFIS

Speed training data		Current training data	
Error	Output	Error	Output
200	5	5	1
175	5	0.1	1
150	5	0.25	0.5
125	5	-0.1	0.1
100	5	0.25	0.4
75	5	0	0.4
50	5	0	0.4
25	5	0	0.4
10	5	0	0.4
8	5	0	0.4
5	5	0	0.4
2	5	0	0.4
1	5	0	0.4
0	5	3.4	1
0	1.15	3	1
0	1.15	2.5	1
0	1.15	2	1
0	1.15	1	1
0	1.15	0.5	1
0	1.15	0.05	0.5
0	1.15	0.1	0.55
200	5	0.2	0.7
175	5	0.25	0.85
150	5	0.15	0.85
125	5	0.1	0.85
100	5	0	0.85
75	5	0	0.85
50	5	0	0.85
25	5	0	0.85
10	5	0	0.85
8	5	0	0.85
5	5	0	0.85
2	5	0	0.85
1	5	0	0.85
0.5	2	0	0.85
0	1.35		
0	1.35		
0	1.35		
0	1.35		

CHECKING DATA FOR ANFIS

Speed checking data		Current checking data	
Error	Output	Error	Output
200	5	5	1
210	5	0.1	1
200	5	0.25	0.5
175	5	-0.1	0.1
150	5	0.25	0.2
125	5	0	0
100	5	0	0
75	5	0	0
50	5	0	0
25	5	0	0
10	5	0	0
5	5	0	0
4	5	0	0
3	5	3.4	1
2	5	3	1
1	5	2.5	1
0.5	4.5	2	1
0	4	1	1
0	3.9	0.5	1
0	3.9	0.05	0.5
0	3.9	0.1	0.55
200	5	0.2	0.7
175	5	0.25	0.7
150	5	0.15	0.7
125	5	0.1	0.7
100	5	0	0.7
75	5	0	0.7
50	5	0	0.7
25	5	0	0.7
10	5	0	0.7
5	5	0	0.7
4	5	0	0.7
2	5	0	0.7
1	5	0	0.7
0.5	4.5		
0.2	4.1		
0.2	4.1		
0.2	4.1		

REAL-TIME RESULTS

Real-time results using ANFIS controllers at different speeds:

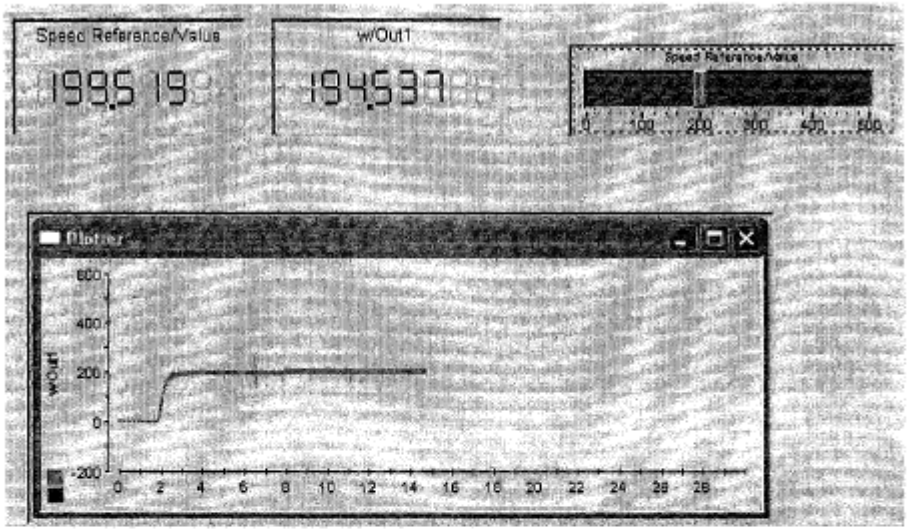

Figure D1. At reference speed = 199.5 rad/sec

When reference speed is given as 199.5 rad/sec, the output is 194.537 rad/sec.

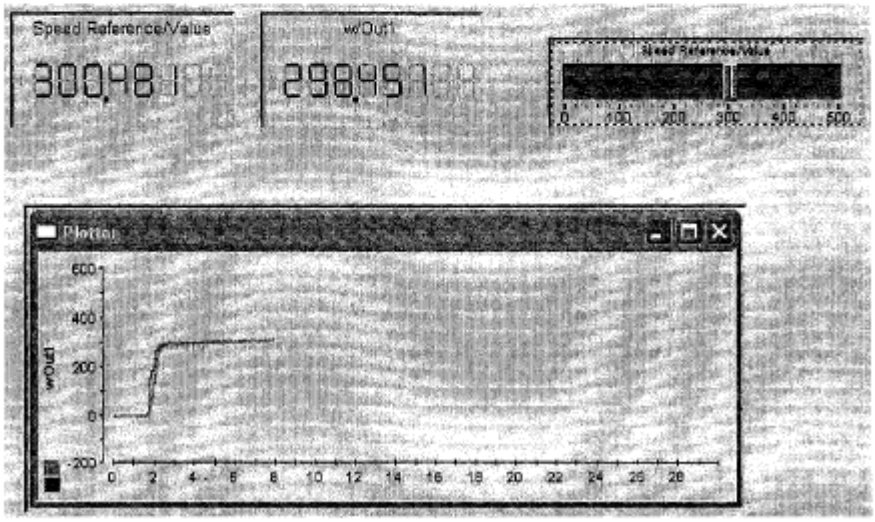

Figure D2. At reference speed = 300.48 rad/sec

When reference speed was set at 300.48 rad/sec, the output was 298.451 rad/sec

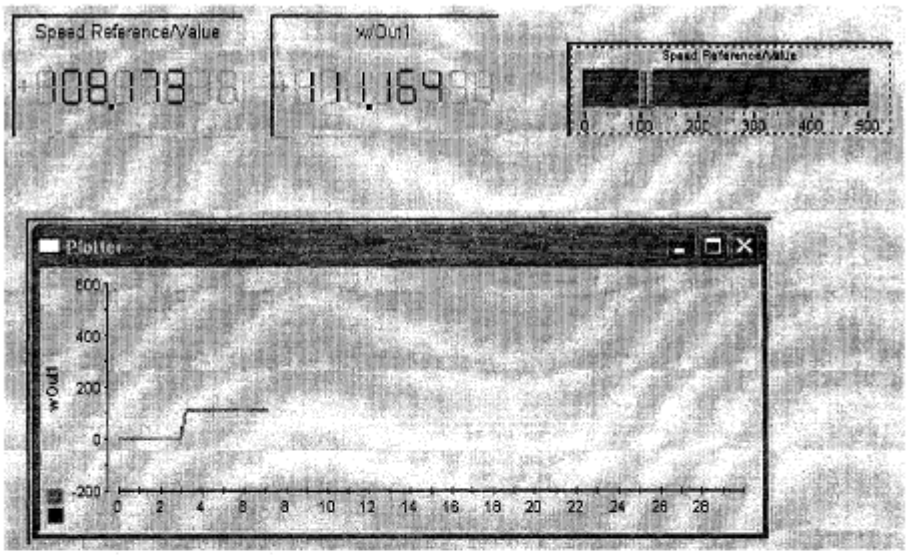

Figure D3. At reference speed = 108.17 rad/sec

When reference speed was set at 108.17 rad/sec and 396.63 rad/sec, the output was 111.164 rad/sec and 396.32 rad/sec respectively.

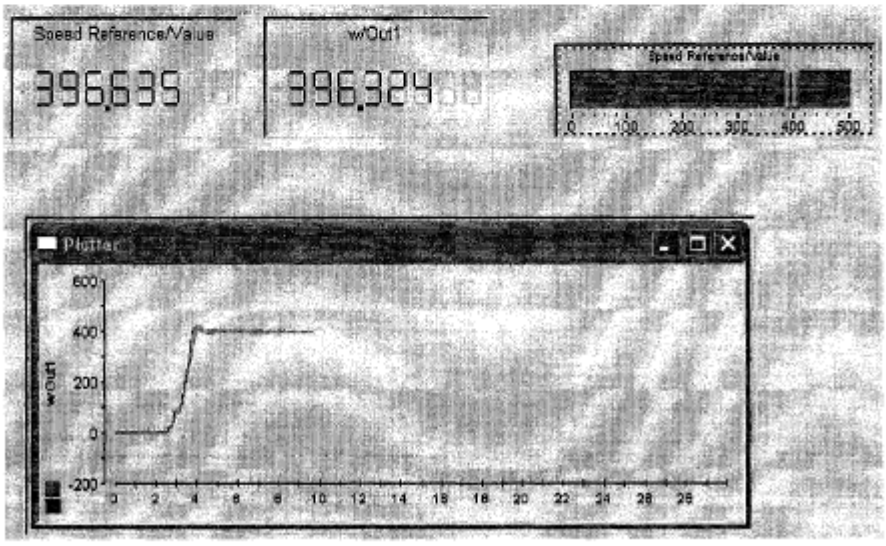

Figure D4. At reference speed = 390.58 rad/sec

Real-time results of using PID controllers at different speeds:

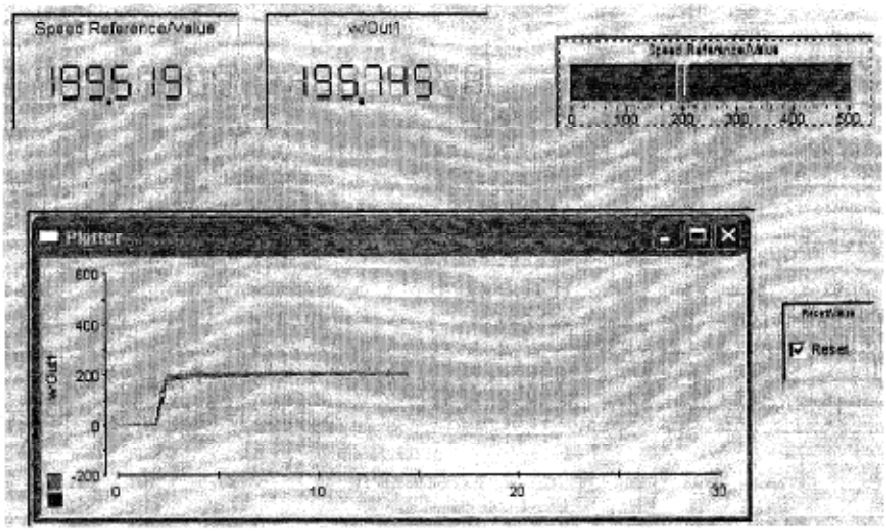

Figure D5. At reference speed = 199.51 rad/sec

When reference speed was set at 199.51 rad/sec and 103.36 rad/sec, the output was 195.74 rad/sec and 103.914 rad/sec respectively.

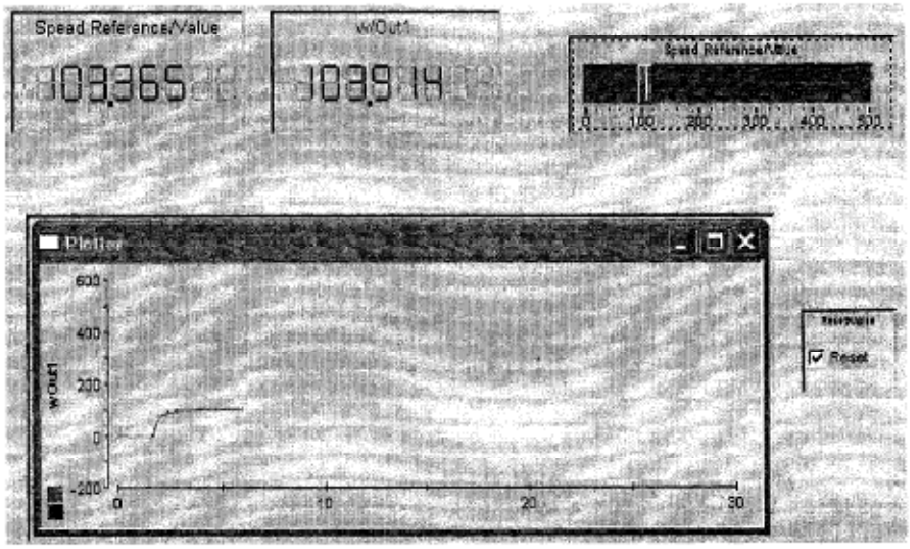

Figure D6. At reference speed = 103.36 rad/sec

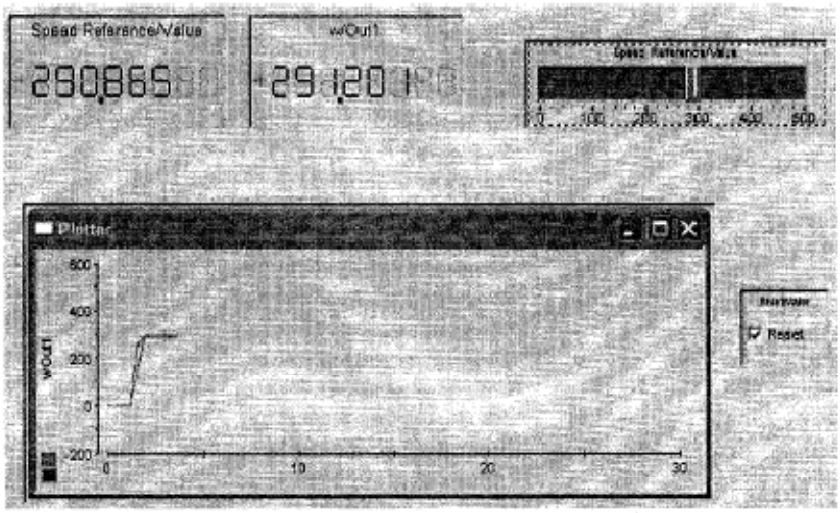

Figure D7. At reference speed = 290.865 rad/sec

When reference speed was set at 290.865 rad/sec and 360.57 rad/sec, the output was 291.20 rad/sec and 358.86 rad/sec respectively.

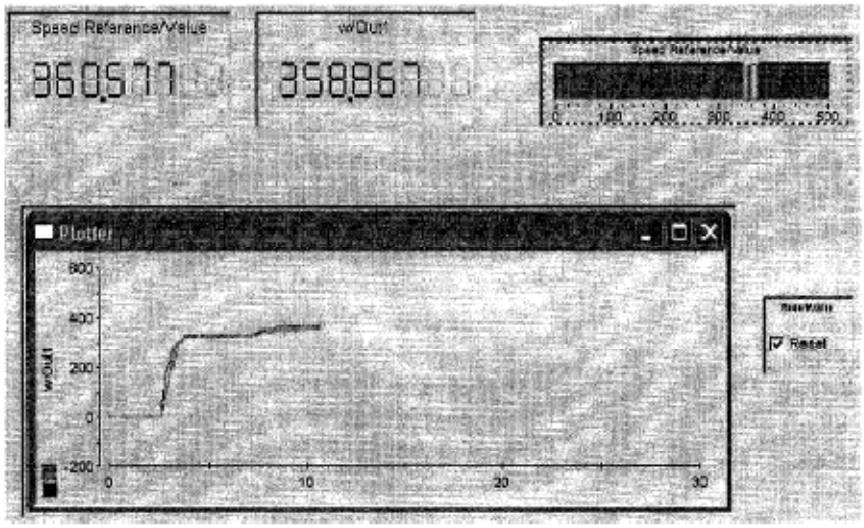

Figure D8. At reference speed = 360.577 rad/sec

Figures D9 and D10 shows the responses for a step change to PID and ANFIS controllers respectively.

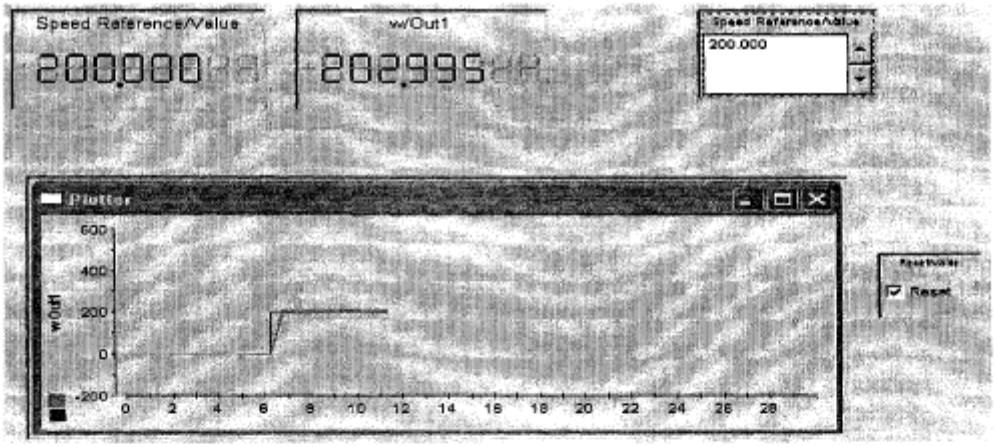

Figure D9. Response of PID at reference speed = 200 rad/sec

(step change from 0 to 200 rad/sec)

When a reference speed of 200 rad/sec was given with a step change from 0 to 200 rad/sec to a PID controller, the output is 202.99 rad/sec. It is observed that the steady state error is ~3 rad/sec and there is a slight delay in the rise time. When the same input is given to ANFIS controller, the output is 202.995 rad/sec. It is observed that the steady state error is ~3 rad/sec while the rise time is good when compared to PID controllers.

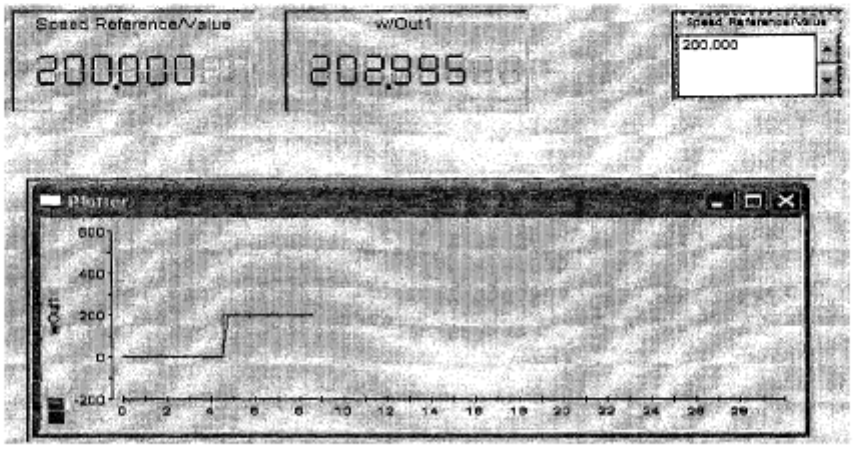

Figure D10. Response of ANFIS at reference speed = 200 rad/sec

(step change from 0 to 200 rad/sec)