مقایسه ی انواع کنترل کننده ها با استفاده از شبکه عصبی:

کنترل کننده ی پیش بین:این کنترلر از یک شبکه ی عصبی برای پیش بینی پاسخ آینده ی دستگاه به ورودی های بالقوه کنترلی استفاده میکند سپس از یک الگوریتم بهینه سازی برای برای محاسبه سیگنال کنترلی برای بهینه سازی پاسخ آینده ی دستگاه استفاده میکند .

شبکه ی عصبی در این مدل به صورت آفلاین آموزش میبیند ولی در نهایت این کترلر به حجم زیای محاسبه ی آنلاین نیاز دارد که به این دلیل که الگوریتم بهینه سازی باید در هر نمونه ی زمانی انجام پذیرد.

NARMA-L2 Control

این کنترلر ساختار ساده ای دارد و بر اساس باز آرایی مدل شبکه عصبی که به صورت آفلاین آموزش میبیند عمل میکند

تنها محاسبه آنلاین این کنترلر گذر رو به جلوی شبکه عصبی در حین کنترل فرایند است . اشکال این شبکه این است که فرم مدل سیستم باید نزدیک فرم کامپونین باشدیا قابل تخمین با سیستمی باشد که فرم کامپونین دارد.

Model Reference Control

مانند کنترلر قبلی تنها محاسبات آنلاین آن مینیمم است هرچند بر خلاف کنترلر قبلی این کنترلر به یک شبکه ی عصبی که به صورت آفلاین آموزش دیده است نیز نیاز دارد

هزینه ی محاسباتی آموزش این شبکه زیاد است چون از الگوریتم پس انتشار خطا استفاده میکند .

ولی مزیت این شبکه این است که برای سیستم های با تنوع بیشتر نسبت به کنترلر قبل کاربرد دارد .

NARMA-L2 (Feedback Linearization) Control

این کنترلر با دو نام شناخت میشود NARMA-L2 و کنترلر خطی سازی فیدبک

در صورتی که سیستم مورد نظر به فرم کاپونین باشد از آن به عنوان کنترلر خطی سازفیدبک یاد میشود و اگر از تقریب سیستم استفاده شود به آن NARMA-L2 یاد میشود

ایده ی اصلی این کنترلر این است که دینامیک سیستم غیر خطی را با حذف عوامل غیر خطی آن به یک سیستم خطی تبدیل کنیم

اولین گام برای استفاذه از این سیستم این است که مدل فرایند مورد مطالعه استخراج شود و شبکه ی عصبی آموزش ببیند تا دینامیک پیش رو سیستم را ارائه کند یک مدل استاندارد که برای ارائه ی مدل شبکه های غیر خطی زمان گسسته استفاده میشود مدل NARMA h که با معادله ی زیر ارائه میشود:

*y**k* + *d*= *N**y**k**y**k* – 1...*y**k* – *n* + 1*u**k**u**k* – 1...*u**k* – *n* + 1

در فاز شناسای سیستم باید شبکه ی عصبی با داده های مطلوب آموزش داده شود تا تابع N تقریب بزند . اگر بخواهیم خروجی یک مقدار مرجع را تعقیب کند قد بعدی این است که کنترلر را به صورت زیر طراحی کنیم

*u**k*= *G**y**k**y**k* – 1...*y**k* – *n* + 1*yr**k* + *d**u**k* – 1...*u**k* – *m* + 1

مشکل استفاده از این کنترلر این است که اگر بخواهیم یک شبکه ی عصبی را آموزش دهیم تا تابع G را تقریب بزند و مربع خطا را مینیمم کند باید از الگوریتم پس انتشار خطا استفاده کنیم که سرعت کنترل کاملا پایین می اید . یک راه حل برای این مشکل این است که از تقریب سیستم به صورت زیر استفاده کنیم

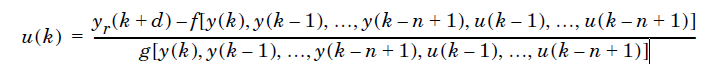
*y**k* + *d*= *f**y**k**y**k* – 1...*y**k* – *n* + 1*u**k* – 1...*u**k* – *m* + 1

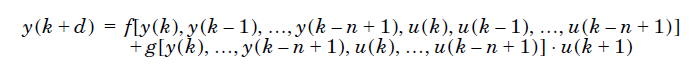
+ *g**y**k**y**k* – 1...*y**k* – *n* + 1*u**k* – 1...*u**k* – *m* + 1\**u**k*

مزیت این روش این است که میتوان سیگنال کنترلی پیدا کرد که نتیجه ی زیر حاصل شود:

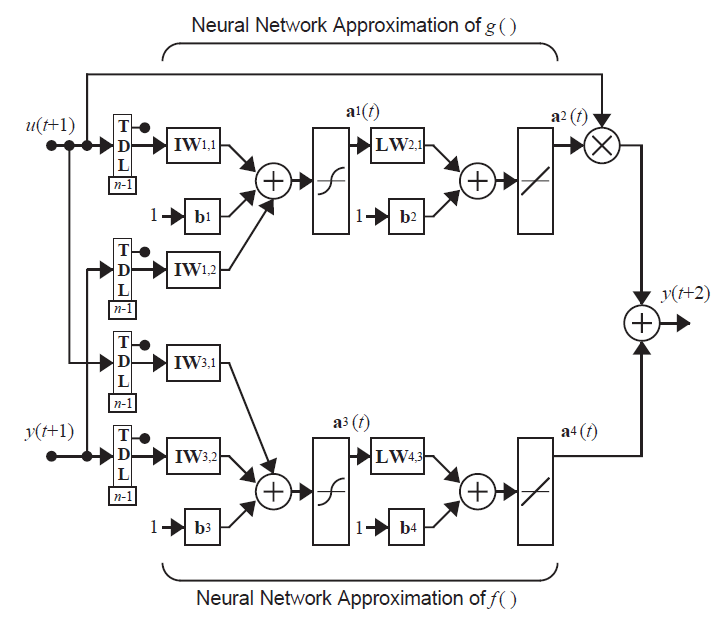
*y**k* + *d*= *yr**k* + *d*

نتیجه ی بدست آمده برای سیگنال کنترلی به صورت زیر است :

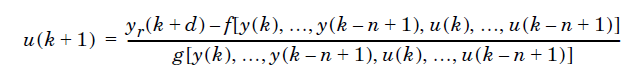




ساختار شبکه ی عصبی در نمودار زیر نشان داده شده است:



در نهایت معادله مورد استفاده به صورت زیر است:



دیاگرام بلوکی فرایند کنترل در شکل زیر نمایش داده میشود: