روشی های نوین برای کنترل میانی موتور های پله ای

ابراهیم بهروزیان نژاد امین جوادی نسب پروین عابدی ساناز نوروزی لرکی داریوش زین العابدینی
Ebrahim_behrouzian@yahoo.com A.javadi62@gmail.com novapar@gmail.com sachli79@yahoo.com abedini84@gmail.com
دانشگاه آزاد واحد شوشتر دانشگاه آزاد واحد دزفول دانشگاه آزاد واحد ماهشهر دانشگاه آزاد واحد ماهشهر دانشگاه آزاد واحد شوشتر

دانشگاه آزاد اسلامی واحد شوشتر—باشگاه پژوهشگران جوان

چکیده – موتور پله ای وسیله پر مصرفی است که پالس های الکتریکی را به حرکت مکانیکی تبدیل می کند . در کاربرد هایی همچون راه انداز های دیسک و چاپگرهای ماتریسی و رباتیک از موتور پله ای برای کنترل موقعیت استفاده شده است در این مقاله روش های جدیدی را برای کنترل موتور های پله ای آورده شده است . البته در این مقاله بیشتر بحث در مورد کنترل موتور های سطح پایین است.

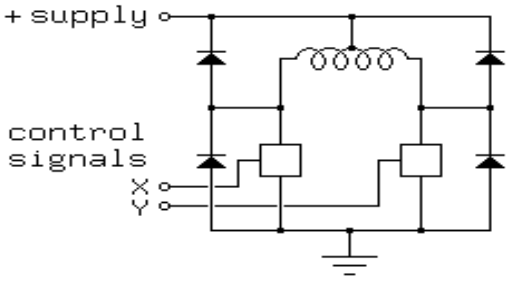
کلید واژه- کنترل میانی.- موتور پله ای- موتور های سطح پایین - کنترل کننده-

۱- مقدمه

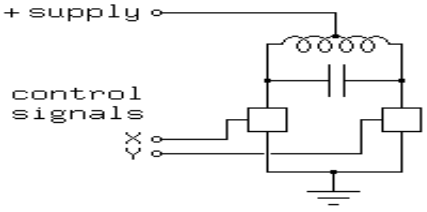
کنترل موتورهای سطح پایین تقریبا مشابه هستند، در یک سطح خلاصه. هر وجه اشتراک تعدادی از درون دادهای منطقی را دارد. بعضی از این درون دادها ممکن است با کنترل مستقیم سوییچ هایی که باز هستند یا بسته استفاده شوند، بقیه ممکن است رمزی شده باشند، درحالی که بقیه ممکن است زیر سیستم هایی از قبیل شباهت به مبدل دیجیتال را در یک وجه اشتراک حرکت های کوچک را کنترل کند.

این حالتهای هر کدام از این درون دادهای منطقی به بردار کنترل برای موتور اشاره می شود، و یک توالی از حالتهای استفاده شده برای چرخاندن موتور به عنوان یک خط سیر کنترل برای موتور شرح داده خواهد شد.

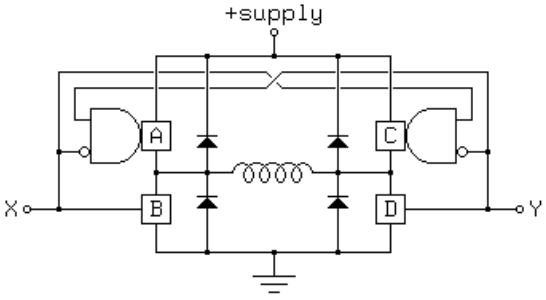
برای مثال، بردار کنترل برای کنترل یک آهن ربای ثابت یا موتور حرکتی مرکب هر مدار کنترل را که در شکل های زیرنشان داده شده اند، استفاده می کند که شامل ۴ قطعه است که ۲ قطعه برای کنترل هر سیم پیچ موتور است.



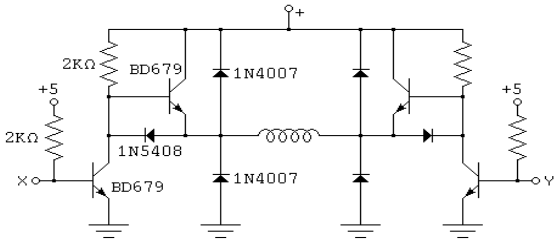
شکل۱



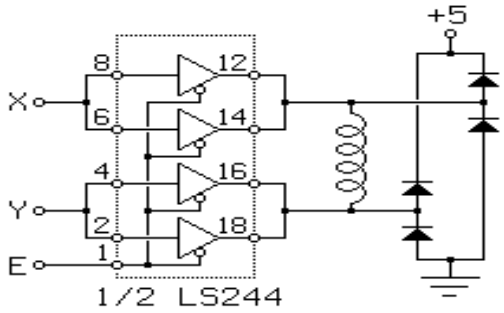
شکل۲



شکل۳



شکل۴



شکل۵

در هر مورد،بردار کنترل می تواند با عنوان $(x_1y_1x_2y_2)$ بیان شود جایی که X_1 و Y_1 کنترل جریان را از طریق سیم پیچی موتور۱ و Y_2,X_2 کنترل جریان را از طریق سیم پیچ۲ انجام می دهند. برای هر وجه اشتراک با این بردار کنترل، خط سیر زیر موتور را از طریق یک چرخه الکتریکی کامل پیش خواهد برد، با استفاده از یک حرکت کامل :

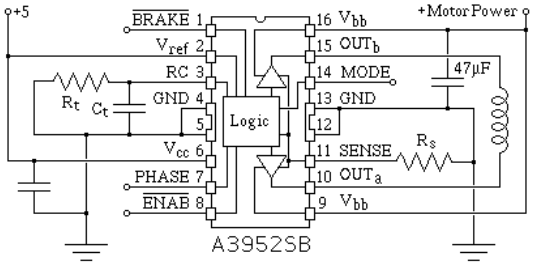
1010
1001
0101
0110
1010

بهمین نحو خط سیر زیر موتور را از طریق همان چرخه الکتریکی به طور نیمه پیش خواهد برد.

1010
1000
1001
0001
0101
0100
0110
0010

1010

کنترل کننده های دیگر بردارهای کنترلی مختلف دارند. برای مثال، خط سیر کنترل برای یک آهن ربای ثابت با موتور مرکب که بوسیله یک جفت از ۳۹۵۲ Allegro کنترل شده اند قطعه هایی که $(B_1E_1P_1M_1B_2E_2P_2M_2)$خواهد شد،



شکل۶

جایی که Mode,Phase, ENABLE, BRAKE, M, B,E,P درون دادهای کنترلی برای هر قطعه هستند. در این مورد، خط سیر کنترل زیر حرکت کامل موتور از طریق چرخه الکتریکی خواهد بود. خط سیر کنترل زیر نصف حرکت همان موتور از طریق یک چرخه الکتریکی خواهد بود.

10001000
10001010
10101010
10101000
10001000

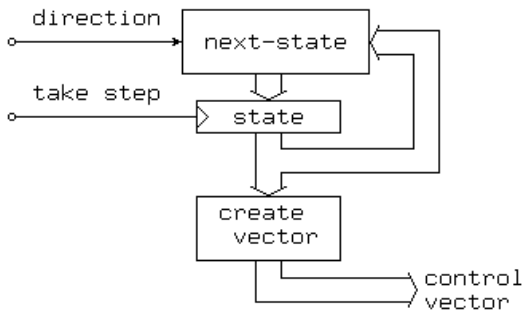
با استفاده از مانع دینامیک برای کنترل شدید هر موقع که سیم پیچ خاموش است.

10001000
10000000
10001010
00001010
10101010
10100000
10101000
00001000
10001000

این یک چیز ارزشمند است، در حالی که مانع دینامیک روی سیم پیچ های مجزا یک روش عالی برای کنترل تشدید در طول عملیات با سرعت پایین، در سرعتهای بالا موتور، گشتاور کاهش پیدا خواهد کرد.
بردارهای کنترلی مورد نیاز برای موتورهای با حرکت کم ، خیلی پیچیده هستند، اما عقیده اساسی به همان شکل باقی می ماند. مشکل ما روبرو شدن با توسعه سیستم های کنترلی سطح بالاتر است که خط سیرهای کنترلی مناسبی را تولید خواهند کرد،موتور یک مرحله ای، نصف حرکت یا حرکت های کوچک حرکت می کند هر زمان که سیستم کنترلی سطح بالاتر به یک حرکت نیاز داشته باشد.

۲- راه حل های سخت افزاری

راه حل های ابتدایی برای این مشکل تولید خط سیرهای کنترلی مناسب برای حرکت موتور همیشه بر اساس ترکیب مستقیم خط سیر کنترلی در سخت افزاراست. بعضی راه حل ها هنوز برای بعضی کاربردها مناسب هستند، اما این روزها، زمانی که وجه اشتراک قطعه های کنترل کننده برای جایگزین کردن سرعت پایین استفاده می شوند و زمانی که بیشترین حرکت عملکردهای موتور سرانجام بوسیله سیستم های کامپیوتر از یک نوع به نوع دیگر کنترل می شوداستفاده از نرم افزار برای اینکار پیشنهاد می شود.
همه راه حل های سخت افزاری برای تولید خط سیر کنترلی بوسیه مدل کلی شرح داده شده در شکل زیر رده بندی می شوند:



A general model for mid-level control

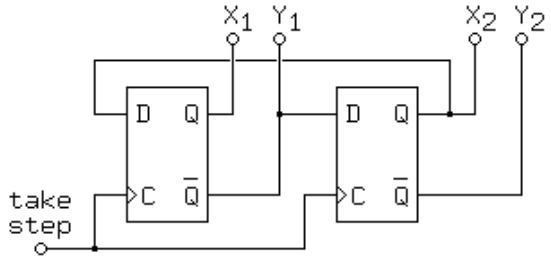
شکل۷

قسمتهای next-state, creat-vector در شکل بالا کاربردهای منطقی ترکیبی هستند، در حای که قسمت state یک ثبات است. بستگی به موقعیت رمزی شده دارد وظیفه creat- vector ممکن است جزئی باشد، برای ایجاد عملکردها، وظیفه next-state همچنین جزئی است، اما در موارد کلی،next- state یک شمارش گر up-down می شود در حالی که creat – vector یک rom می شود که توالی از بردارهای حالت مورد نیاز را نگه می دارد .

این یک چیز با ارزش است که بعضی سیستم های کنترلی حرکت موتور شامل درون دادهای اضافی به سیستم کنترلی سطح متوسط می شوند.افزایش های معمول درون دادهای half –step , brake را شامل می شود. کنترل مانع در حالت حرکت کامل بیهوده و بی معنی است اما در حالت نصف حرکت استفاده می شود. در آخرین حالت، برای سیم پیچ های بدون استفاده باید برای کارآیی باز باقی بماند.سرانجام، بعضی سیستم های کنترلی شامل یک درون رها کردن هستند که انرژی را از همه سیم پیچ ها رفع می کند در این حالت، اگر مانع اثبات شود، همه سیم پیچ ها کوتاه می کند.

۳- مثالهای عملی

ستاره شناسان مکرراً به موتورهای خیلی آهسته برای چرخاندن تلسکوپهایشان نیاز دارند،و در سالهای اخیر، حرکت موتورها در موتورهای هماهنگ و نظم ابزار برای بسیاری عملکردها واقع شده اند، به ویزه برای در انبار یا نصب scotch، یک گروه از دوربینهای خیلی ساده نصب شده در عکس برداری از ستارگان استفاده می شوند. برای این عملکرد، مدارها به سادگی که در شکل زیر نشان داده شده هستند.

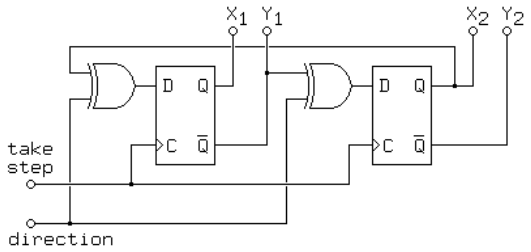


شکل۸

مدار نشان داده شده در شکل بالا فقط در یک جهت عمل می کند،تولید علامت های مورد نیاز برای چرخش یک آهن ربای ثابت یا موتور مرکب در یک حرکت کامل برای هر ضربه روی درون داد حرکت داده شده.بر حسب مدل کلی در شکل ۱-۶،کاربردهای next – state و creat –

vectore جزئی هستند و به هیچ منطقی برای تولید هر flipflop در حالت ثابت برون دادهایی در هر دو شکل کامل و مستقیم دارد.

این مدار برای عملکرد دو جهتی مستقیم به سمت جلو است، همانطور که در شکل زیر شرح داده شده است :



شکل۹

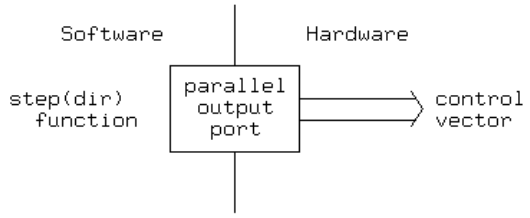
مدار نشان داده شدده در شکل ۹، دو محدود کننده یا مخزن برای تخمین دو نوع مختلف از کارکرد – next state را استفاده می کنیم، بستگی به مقدار درون داد جهت دارد. هر flip- flop دو برون داد inverted و non-inverted را به جهان ارایه می دهداین کار اجازه می دهد که یک مدار هم نیرو بوسیله جانشین سازی یک سوییچ دو قطبی برای محدودکننده یا مخزن درست شود، و مدار هم نیروی دیگر از این بوسیله جانشین سازی جفت از تسهیم کننده های ۲ درون داد ۱یرون داد برای سوییچ می تواند بدست بیاورد.

تعدادی از سازنده های مدارهای کنترل حرکت موتور مدارهای جامع و یکپارچه ای درست می کنند که شامل منطق پیچیده تری است که به هر دو حالت حرکت کامل و نیمه اجازه می دهد. برای مثال Motorola و قطعه MC3479 شامل یک H-bridge ۱۶ ولتی،ma ۳۵۰ است و کنترل منطقی با حرکت، جهت و حالت درون دادها برای کنترل نیمه حرکت.

SGS – Thampson قطعه L۲۹۷ فقط شامل کنترل سطح متوسط برای کنترل حرکت کامل یا نصف حرکت از یک جفت H-bridge استفاده شده برای کنترل یک آهن ربای ثابت یا موتورهایی با مقاومت مغناطیسی متغییر است. این قطعهه مخصوصاً برای کنترل L298dual ، H-bridge ۲ آمپری است و علاوه بر این برای کنترل سطح متوسط، شامل یک one- shots است و مقایسه کننده های مورد نیاز برای استفاده این H-bridge به عنوان یک جریان محدود کننده تقسیم کننده است.

۴- راه حل های نرم افزاری

اگر یک پردازشگر کوچک وجه اشتراک کنترل کننده را یا سیستم کامپیوتری دیگر را برای کنترل یک حرکت موتور استفاده کند، کامپیوتر می تواند بردار کنترل را در نرم افزار تولید کند و آن را به موتور از طریق یک برون داد همزمان ارایه دهد.



A general model for mid-level control

شکل ۱۰

در این مدل متمرکز شده روی نرم افزار، مشکل اساس کنترل سطح متوسط برای چگونگی کارکرد حرکت اجرا شده کاهش می یابد.هر درخواستی برای حرکت (d) باعث می شود بردار کنترل پیشرفت کند برای خط سیر کنترل در جهت خاصی بوسیله dir . درخواست حرکت (۱+) باعث می شود موتور به طرف جلو حرکت کند، در حالی که درخواست حرکت (۱-) باعث می شود یک مرحله به عقب برگردد.

خط سیر کنترلی برای هر موتور می تواند به عنوان یک نظم دایره ای از بردارهای کنترل توصیف شود. خط سیر ساده یک حرکت کامل در مقدمه داده شده بود می تواند به صورت زیر ارائه شود:

۴ عنصر نظم و آرایش جایی که a و t

$t[0] = 10$
$t[1] = 9$
$t[2] = 6$
$t[3] = 5$

به همین نحو، خط سیر یک حرکت تیمه برای همان موتور وجه اشتراک خواهد داشت:

$t[0] = 10$
$t[1] = 8$
$t[2] = 9$
$t[3] = 1$
$t[4] = 5$
$t[5] = 4$
$t[6] = 6$
$t[7] = 2$

تعریفهای مشابه :فقط در اندازه آرایش خط سیر و مقدارهای هر ورودی متفاوت است، برای توصیف یک چرخه کامل از خط سیر کنترل برای حرکت موتور کافی خواهد بود!
خود روال حرکت به هر روشی در بردار کنترل بستگی ندارد.
برای هر مثال بالا، روال حرکت به این شکل است، p ،یک عدد صحیح متغیر، از صفر شروع می شود.
یک عملکردی از متغیر مستقل d
جایی که d باید با ۱+ یا ۱- باشد.
بدون مقدار بازگشت داده می شود.
در اینجا، ما فرض می کنیم که عملکرد برون داد یک بردار کنترل را به موتور می فرستد. جزئیات این عملکرد به کامپیوتر، وجه اشتراک استفاده شده و سیستم عملکردی، اگر وجود داشته باشد بستگی خواهند داشت. شبه برنامه بالا همچنین یک عملکرد مقیاسی را که مقیاس نظم حفظ شده یک چرخه از خط سیر را منعکس می کند را در نظر می گیرد، چگونه این تغییر از یک زبان برنامه ریزی شده به زبان دیگر تغییر می کند.
زمانی که رمز بالا را به زبان های برنامه ریزی شده به زبان دیگری تغییر می کند.
زمانی که رمز بالا را به زبان های برنامه ریزی شده واقعی ترجمه می کنیم، استفاده ساده از اپراتور می تواند به ندرت حفظ شد. ریاضی دانان به طور کلی بحث می کنند که ۰
$y > x \pmod{y} \geq 0$
اما برنامه ریزان زبانها برنامه ریزی شده مکرراً از این تعریف دورمی شوند.زمانی که x منفی است. در نتیجه، در زبانهای از قبیل c ، c + + ، javs و pascal ، باید به اجتناب مقدارهای منفی در سمت راست mod اپراتور توجه شود.
مثالهای عملی ساده
در ادامه رمز c ۳. جنبه مقاومت مغناطیسی متغییر موتور را که به بردار کنترل خودش از سه قطعه مهم پورت موازی از یک سیستم unix یاlinux داده می شود را کنترل خواهد کرد، با این فرض که پورت موازی در حالت صحیح استفاده از فایل pp باز می شود.

```
#define STEPS 3
  int t[STEPS] = { 1, 2, 4 };

  int p = 0;

  void step( int d )
  {
    char c;
    p = (p + STEPS + d) % STEPS;

     /* output t[p] using low-level Unix I/O
*/
    c = t[p];
    write( pp, &c, 1 );
  }
```

دوباره نویسی چند خط اول رمز بالا به ما اجازه می دهد برای انتقال آن برای استفاده با یک آهنربای ثابت یا موتور با مقاومت متغییر در حالت نیمه حرکت ۴ قطعه مهم پورت موازی عمل کنیم

```
#define STEPS 8
  int t[STEPS] = { 10, 8, 9, 1, 5, 4, 6, 2 };
```

۵- یک طرح ناشی از هدف

مسائل خیلی پیچیده تر می شوند اگر موتورهای متعدد در آنها استفاده شودً اگر چه راه حل های ad-hoc می توانند معمولی باشند، یک روش اصولی مناسب این است، و حتی در زبانهایی بدون حمایت برای روش شناسی ناشی از هدف، ساده ترین راه برای راه حل های مطلوب در اصطلاح هایی ناشی از هدف است.
بعد از مقدماتی ، که ممکن است به بسیاری جزئیات سطح پایین بستگی داشته باشد، نرم افزار سطح بالا در چگونگی کار کردن موتور جالب نیست. بنابراین، برای هر هدف موتور، وجه اشتراک های آشکار فقط در عملکرد حرکت مورد نیاز است به طور قابل توجهی از جزئیات چگونگی کارکرد برای دیگرقسمتها متفاوت باشد. رمز زیر در java داده شده است، ترجمه به c++ ، 67 simula یا زبان های ناشی از هدف دیگر باید مستقیم باشد:

```
abstract class StepMotor
  {
    public abstract void step(int d);
       // step the motor in direction d (+1 or
-1)
  }
```

این یک گروه عینی است: یعنی، اهداف این گروه نمی توانند سریعاً انجام شوند زیرا ما باید همه جزئیات چگونگی موتور یا وجه اشتراک کارها را مشخص کنیم. هر وجه اشتراک حرکت مفید موتور باید بوسیله یک گستردگی یا طبقه فرعی این گروه عینی حمایت شود. گروه زیر این را شرح می دهد، ترکیب رمز در این قسمت بالا بحث می شود:

```
class UniversalStepMotor extends StepMotor
  {
    private int s;          // the size of the trajectory
    private int[] t;        // the trajectory for this motor
    private int p;          // motor's position in trajectory
    private OutputStream o;// the output port to use

    public void step(int d)
      // step the motor in direction d (+1 or -1)
    {
      p = (p + s + d) % s;
      o.write( t[p] );
    }

    public UniversalStepMotor( int[] table, OutputStream out )
       // initializer
    {
       s = table.length;
      t = table;
      p = 0;
      o = out;

      o.write( t[p] );
    }
  }
```

رمز بالا فرض می کند که وجه اشتراک برای موتورها در دسترس هستند از طریق جریان برون داد رمز نوع java.io.output stream است، رمزjava زیر یک هدف universalstepmotor برای یک موتور با مقاومت مغناطیسی متغییر phase -۳ است:

```
StepMotor m = UniversalStepMotor(
    new int[] {4,2,1}, // step table for the motor
    MotorPort          // OutputStream for the motor
  );
```

با این بیان و قالب بندی در اینجا، درخواست m-step(1) یک حرکت موتور را خواهد چرخاند. برای بسیاری از عملکردها، گروه stepmotor با یک روش reset نیاز خواهد داشت، برای شروع دوباره هدف موتور و وجه اشتراک موتور استفاده می شود. این روش احتمالاً به طور مستقیم در تعریف گروه step motor زمانی که با اضافه کردن آن بوسیله گسترش گروه مواجه می شوند.این روش باید به عنوان اولین مرحله در جبران حذف خطا در سطح های بالاتر در درجه بندی کنترل نامیده شود.

زمانی که اهداف انجام نمی شوند

برای برنامه ریزان که نمی توانند طرح ناشی از هدف را استفاده کنند، به هر دلیل، مثال زیر یک طرح سطح متوسط مناسب را شرح می دهد که از استفاده از چنین مشخصاتی جلوگیری می کند. استفاده از pascal ، انواع زیر می توانند برای توصیف هر موتور استفاده شوند:

```
type
    direction = -1 .. +1;

    trajectory = array [0 .. MaxTrajectorySize] of int;

    StepMotor = record
      s: integer;    { size of this motor's trajectory }
      t: trajectory;
      p: integer;    { this motor's position in trajectory }
      o: port;       { the output port to use for this motor }
    end { record };
```

نوع ثبت stepmotor در اینجا دقیقاً مطابق با هدف گروه universalstepmotor که در بالای java تعریف شده بود. نظم خط سیر صریحاً نامگذاری شده به طوریکه یک

خط سیر ممکن است، به عنوان یک پارامتر رسمی، برای یک روش ابتدایی برای ثبت های این نوع قرار داده شود. بخش فرعی direction به مکانیسم های محدود بخش فرعی برای بررسی درستی پارامترها برای عملکرد حرکت اجازه می دهد، یک مشخصه مطلوب که از دست داده شده اند در زبان ها از c نزول می کند.

با تعاریف داده شده بالا، ما می توانیم حالا یک هدف کلی روش حرکت را بسازیم.

```
procedure step(var m: StepMotor, d: direction);
    { step motor m in direction d }
  begin
    with m do begin
      p = (p + s + d) mod s;
      output( o, t[p] );
    }
  end { step };
```

در همه موارد بالا، ما فرض می کنیم که متغییرهای نوع port برای پورت های برون داد مطلوب استفاده شده اند که موتورها ممکن است متصل شوند، و روش برون دادهای out put یک بردار به پورت نشان داده شده، است.

در تأخیر، در زبانهای برنامه ریزی شده رها شده با حمایت مستقیم برای منشأ هدف، ما باید عددنویسی را تغییر دهیم:

```
motor.step( dir );
to:
  step( motor, dir );
```

این یک مشکل اصلی در یک سیستم نیست، جایی که فقط یک نوع از وجه اشتراک موتور وجود دارد، امازمانی که چند نوع موتور وجود داشته باشد، برای مدل ناشی از هدف حمایت است مفید خواهد بود. اگر باید در این زبان مانند پاسکال انجام شود، اگهی برای نوع stepmotore باید اصلاح شوود برای اینکه به همه انواع وجه اشتراک های موتور اجازه دهد، برای مثال ، به وسیله تحریم آن برای اینکه یک ثبت متغییر باشد، و روش step باید برای بررسی نوع موتور و عملکرد مناسب اصلاح شود.

۶- مراجع

[1] Youngju Lee, Y.B. Shtessel ''Comparison of a feedback linearization controller and sliding mode controllers for a permanent magnet stepper motor'' IEEE-28th Southeastern Symposium on System Theory (SSST '96) March 31 - April 02, 1996, pp. 258-262

[2] Changsheng Li Elbuluk, M. Dept. of Electr. Eng., Akron Univ., OH, USA;" IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference "

[3] Andreescu G. D.; Popa A.; Spilca A " Two Sliding Mode Based Observers for Sensorless Control of PMSM Drives "