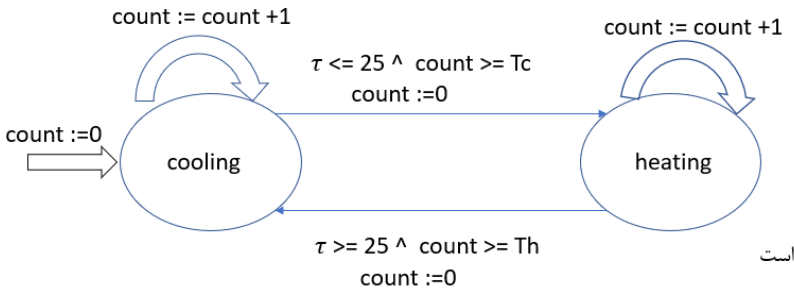
گزارش تکلیف ۲

سید عباس میرقاسمی(۹۷۲۴۳۰۶۸)

سید محمد رضا حسینی(۹۷۲۴۳۱۲۹)

۱-



تحریک شونده با رویداد است

۲- تعداد استیت های ما در کل با فرمول مقابل اندازه گیری میشود: تعداد استیت ها * تعداد حالات متغیر ها

در این جا ما ۳ استیت داریم و تعداد حالات متغیر n ما بی نهایت می باشد. پس در کل فضای استیت بی نهایت می باشد ولی تعداد استیتی که قابل دست یافتن می باشد به شکل زیر می باشد:

۱۶ حالت برایش اتفاق میافتد:

s1/-1|s1/0|s1/1|s1/2|s1/3|s1/4|s2/0|s2/1|s2/2|s2/3|s2/4|s3/1|s3/2|s3/3|s3/4|s3/5

که به ترتیب زیر تولید شده است:

S1/0, s1/1,s2/3,s3/4,

s1/3,s2/0,s3/1,

s1/4,s2/1,s3/2,

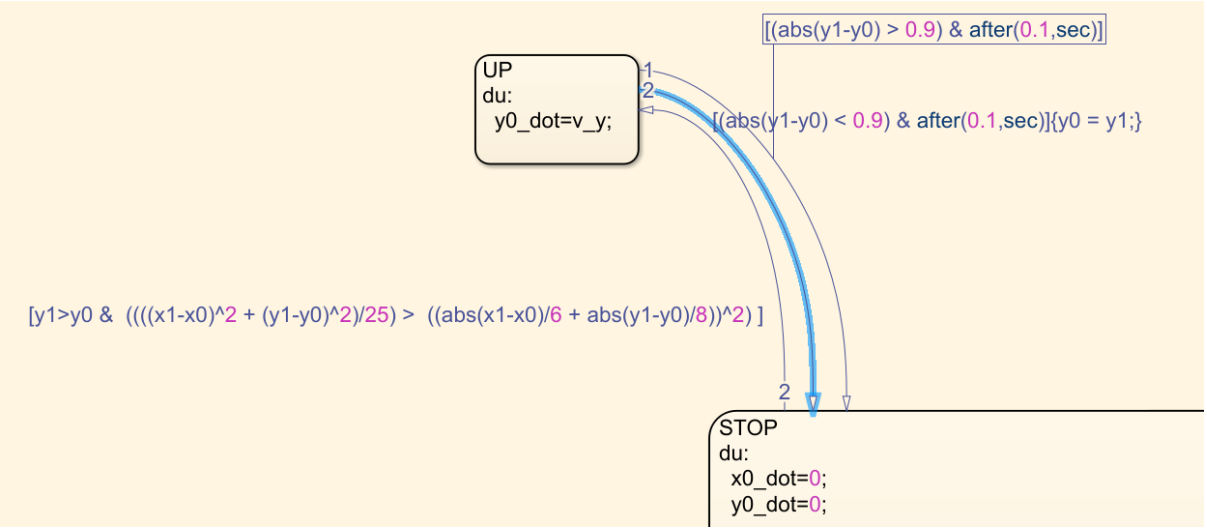
s1/-1,s2/1,s3/2,

s1/2,s2/4,s3/5,

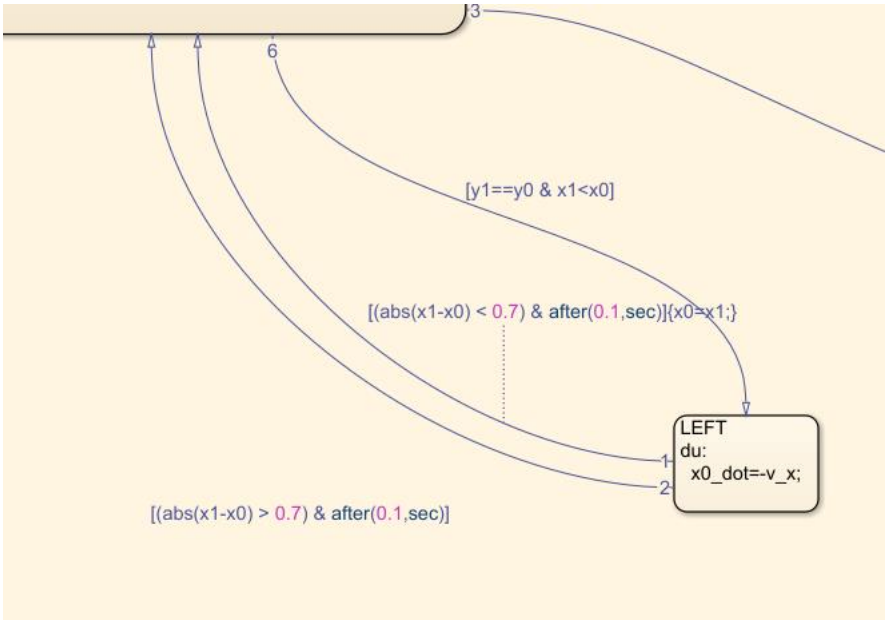
۳- گزارش بخش سیمولینک:

با توجه به صورت سوال، اول ۶ استیت حرکتی تعریف کردیم:

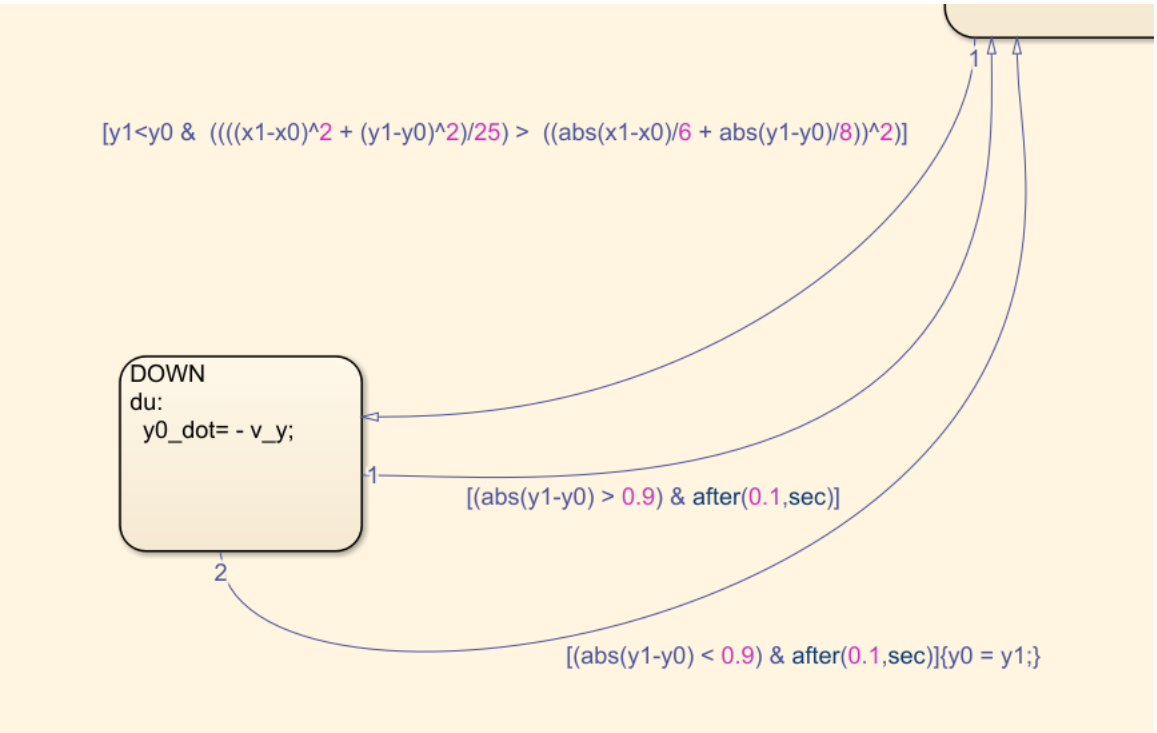
۱- حرکت به بالا



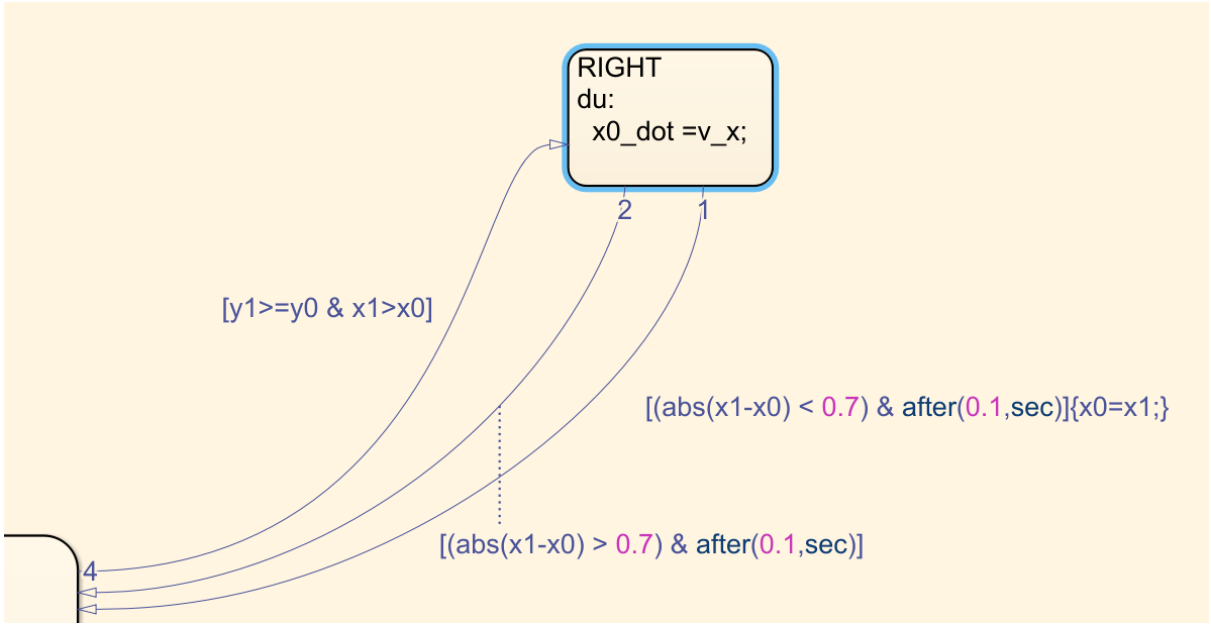
۲- حرکت به چپ



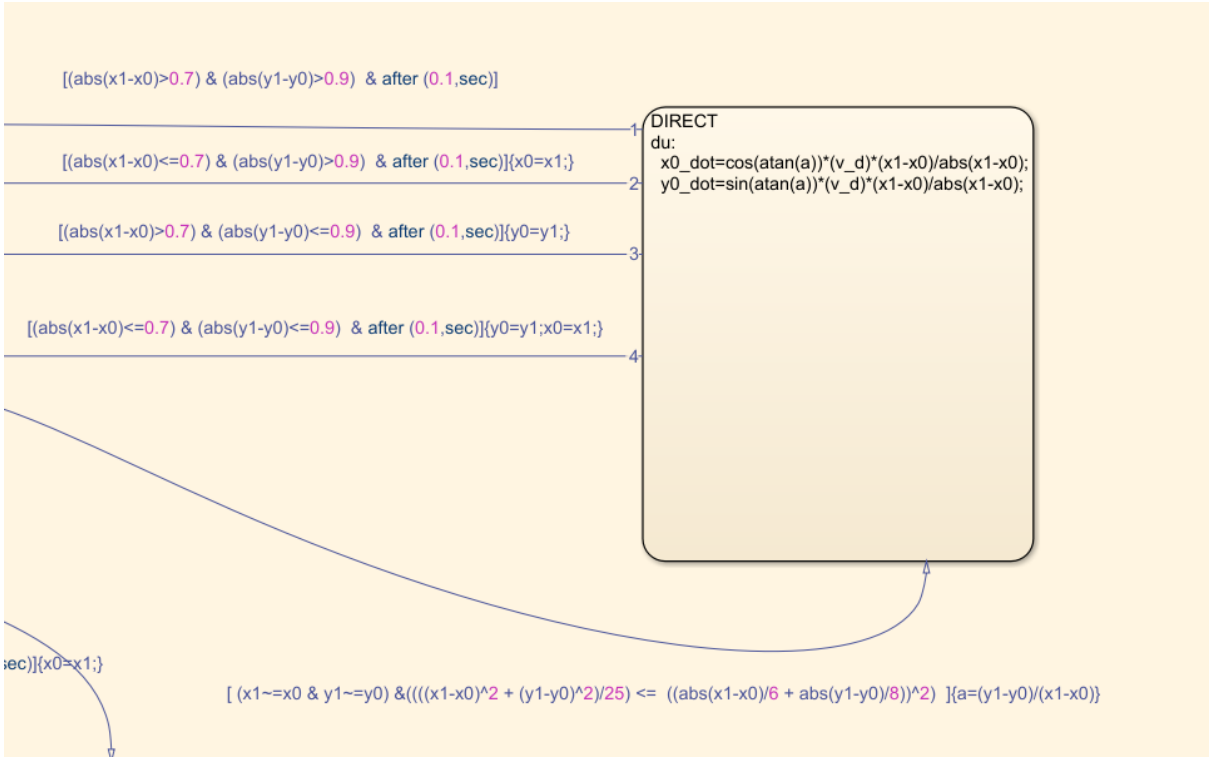
۳- حرکت به پایین



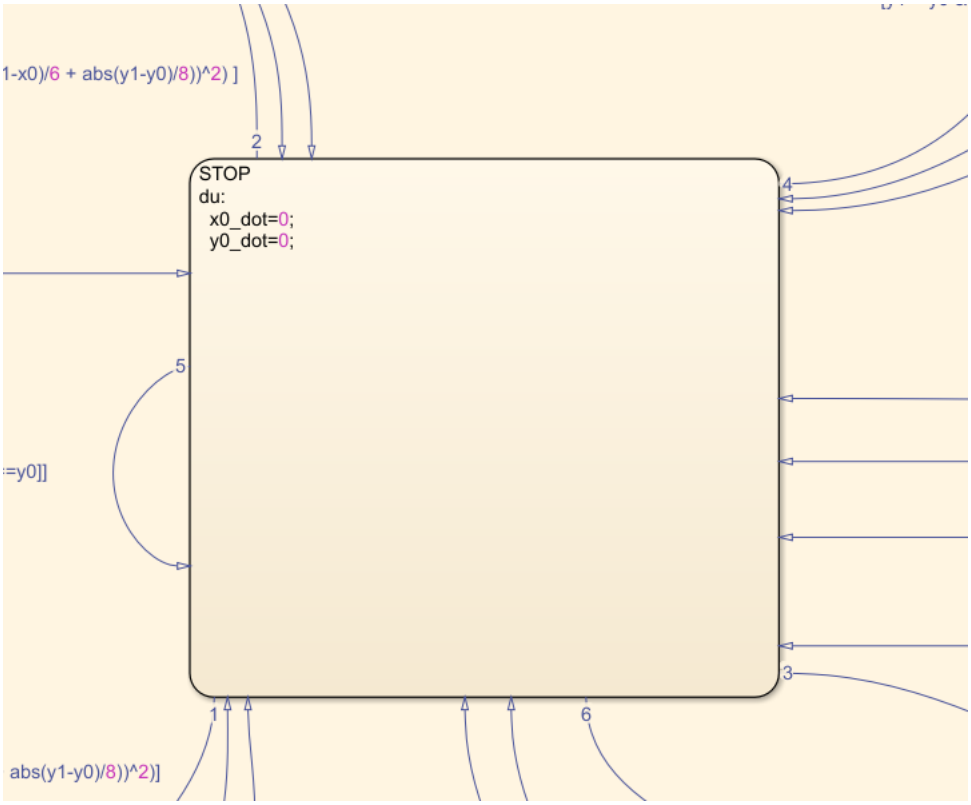
۴- حرکت به راست



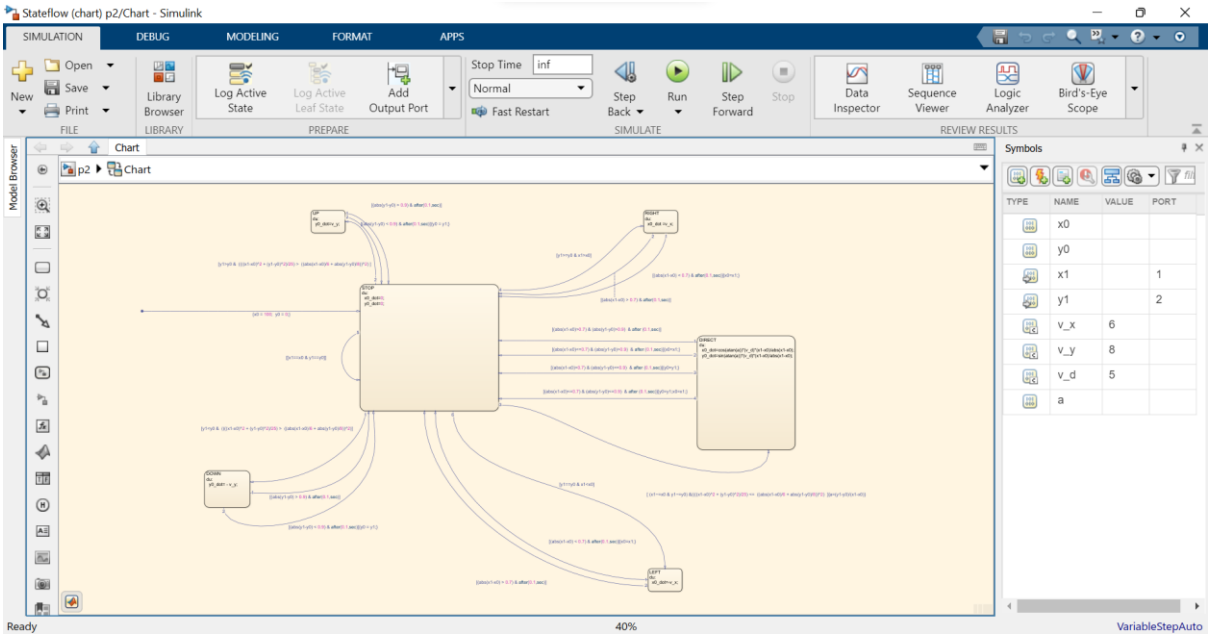
۵- حرکت اریب



۶- ایستادن



در نهایت تصویر کلی از ربات مورد نظرمون:



چند نکته:

- برای ایجاد هرگونه "گام"، "دفعه"، "قدم" از تابع after در واحد ۰.۱ ثانیه استفاده کردیم.
- با توجه به این "گام" رو مقداری بزرگ انتخاب شده است؛ برای این که کد به مشکل نخورد، فاصله های از یک حدی کمتر به صورت مستقیم هندل شده است: برای x فاصله های کمتر از ۰.۷ و برای y فاصله های کمتر از ۰.۹
- بدیهی است برای کمتر کردن خطا میتوان "گام" های زمانی را کوتاه تر کرد.
- برای فاصله مستقیم از فرمول فیثاغورث استفاده شده است.
- برای محاسبه سرعت و موقعیت مکانی در استیت مستقیم از فرمول زیر استفاده شده است:

```
DIRECT
du:
  x0_dot=cos(atan(a))*(v_d)*(x1-x0)/abs(x1-x0);
  y0_dot=sin(atan(a))*(v_d)*(x1-x0)/abs(x1-x0);
```