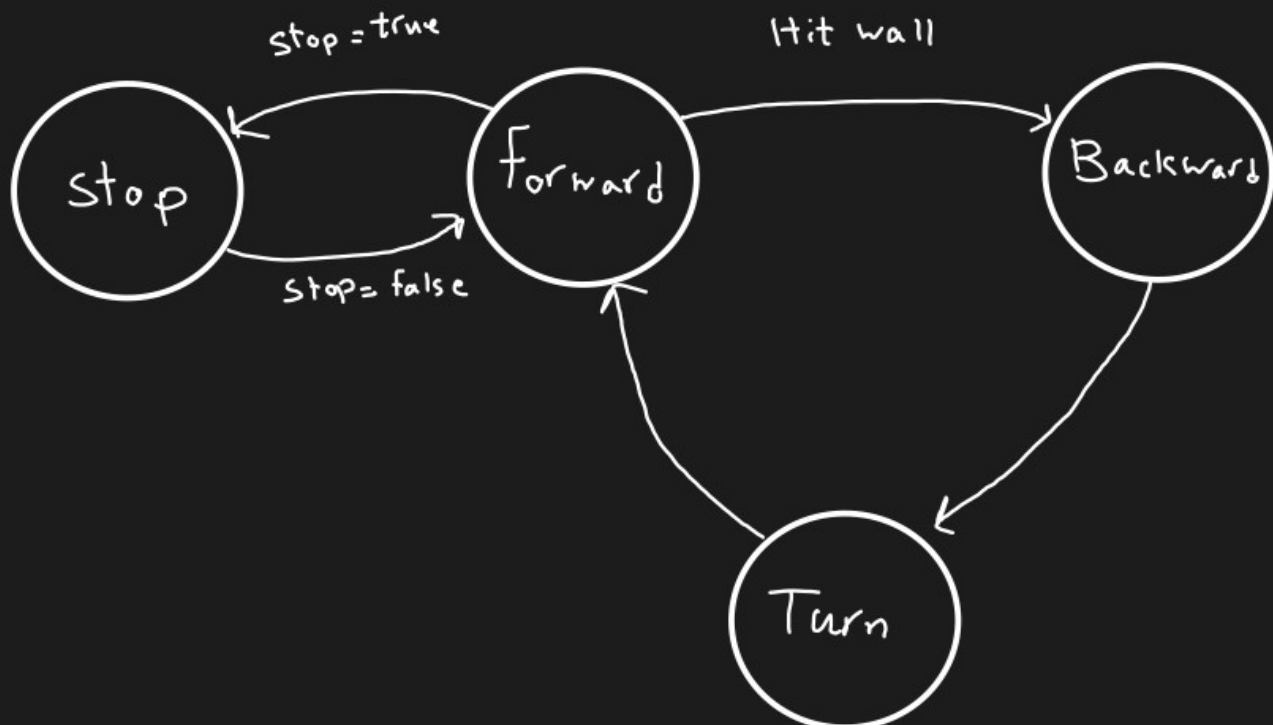


رفتار هوشمند با ماشین‌های حالت محدود

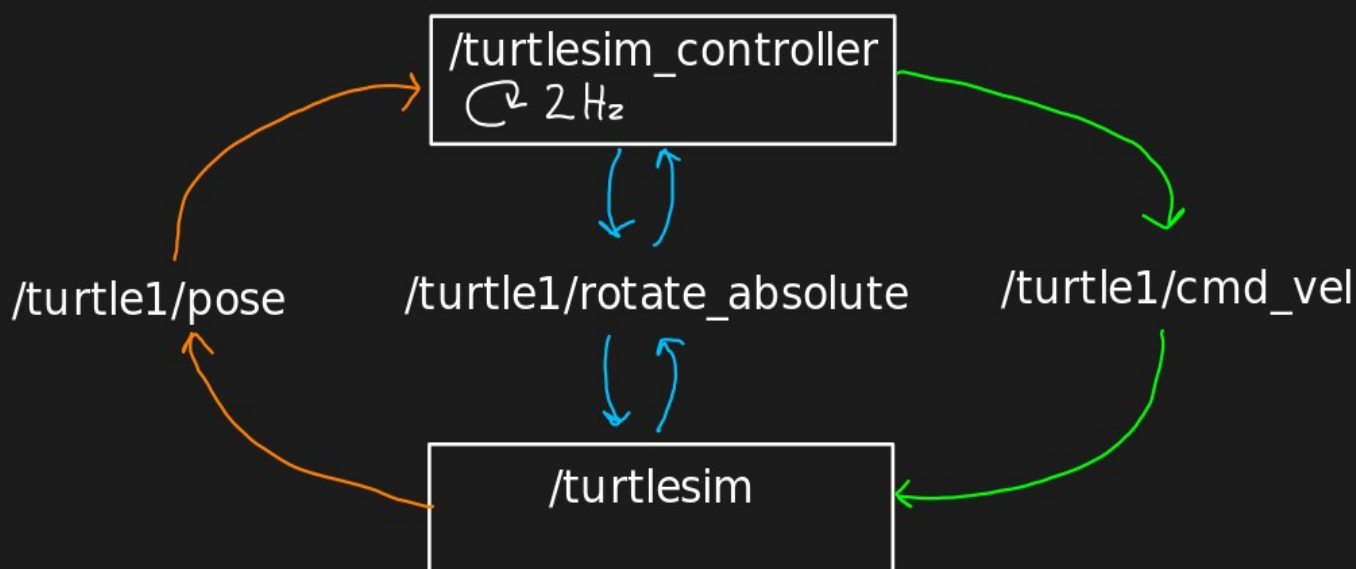
در این آزمایش قصد داریم ساده ترین نوع یک ربات هوشمند را با استفاده از ماشین‌های حالت محدود یا FSM شبیه سازی کنیم. FSM یک مدل محاسباتی ریاضی است که از حالت ها و انتقال ها تشکیل شده و می‌توانیم از آن برای تعریف ساده رفتار یک ربات استفاده کنیم.

یک ربات تا زمانی که در یک حالت است به پردازش و تولید خروجی ادامه می دهد و وقتی شرط یک انتقال برآورده شود و به حالت هدف این انتقال می‌رود. به کارگیری FSM می تواند پیچیدگی حل یک مسئله را زمانی که رفتارهای ساده را اجرا می کنیم به میزان قابل توجهی کاهش دهد.

یک پکیج ROS با نام turtlesim_controller و نود ای با همین نام و با زبان دلخواه طراحی کنید که رفتار یک لاکپشت در turtlesim را با توجه به FSM زیر تعیین می‌کند.



این نود با فرکانس ۲ هرتز فرمان‌های کنترلی خود را به نود `/turtlesim` می‌فرستد. یک پارامتر با نام `stop` دارد که به طور پیش‌فرض `false` است و اگر مقدار آن به `true` تغییر پیدا کند، کنترلر باید حالت خود را به `Stop` ببرد و ماشین حالت را در آن حالت نگه دارد و لاکپشت متوقف شود. در صورت برخورد لاکپشت به دیوار، کنترلر به حالت `Backward` می‌رود و برای تقریباً ۲ ثانیه به عقب حرکت می‌کند و سپس از اکشن `/turtle1/rotate_absolute` استفاده کند تا به طور رندوم لاکپشت را در جهتی بچرخاند و فیدبک مربوط به آن را چاپ کند و بعد به حالت `Forward` باز می‌گردد. ارتباطات نودها در شکل زیر آمده است:



در طی اجرای اگر پارامتر `stop` تغییر کرد، ربات باید به حالت مناسب برود و رفتار درست را انجام دهد. حالت‌های گفته شده باید با نام مناسب در کد وجود داشته باشند و انتقال‌ها از یک حالت به حالت بعد واضح باشند.

در صورت نیاز می‌توانید از کد مربوط به [turtlesim_teleop_key](#) و کد خود [turtlesim](#) کمک بگیرید.

تحويل

از اجرای برنامه در کنار تمام ترمینال‌ها اسکرین شات گرفته و در یک فایل PDF همراه با نام و نام خانوادگی و شماره دانشجویی قرار دهید. نهایتاً فایل PDF را همراه با کد برنامه در یک فایل zip قرار داده و آپلود کنید.