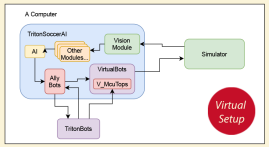
TritonsRCSC

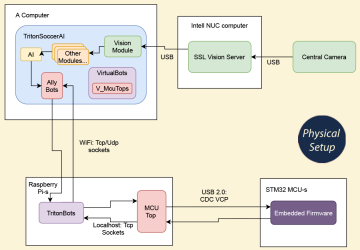
University of California, San Diego

Software

Our Software is highly modularized and distributed, which could adapt to the different setup demands between running in the virtual environment and running with the physical robots. Each module in our software runs concurrently using dependencies such as Java Future, Concurrent, or C++ boost ASIO, which conforms to the high cohesion and loose coupling software design principle. In our new simplified design, our software mainly consists of an AI software written in Java named TritonSoccerAI running the AI program, and a number of embedded software programs named TritonBot written in C++ running on each robot. Similar to the league software, most of the inter-program communications use IP socket + protobuf.

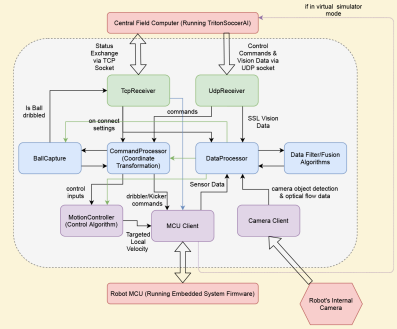
General Setup

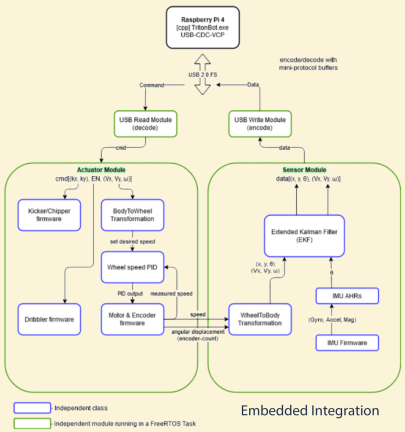




Embedded Software

Our Embedded Software, or software running on each robot, basically follows a distributed computing model in which the TritonBot program handles the computational-heavy algorithms and multithreaded structure, while the stm32 embedded firmware only involves driving the actuators with the raw outputs and getting the mostly raw sensor feedbacks.

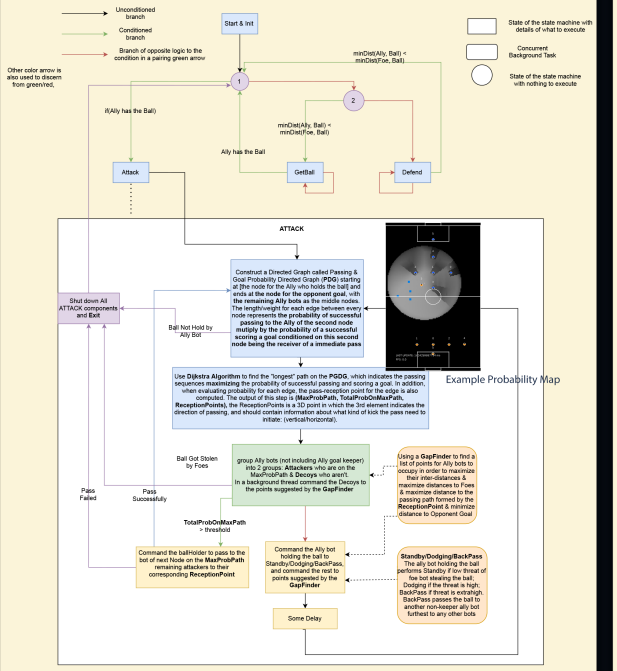




Embedded Integration

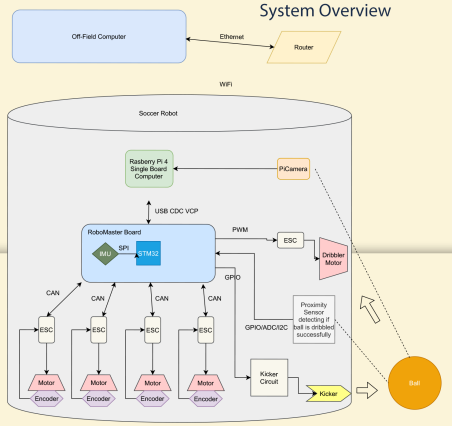
TritonSoccerAI

The TritonSoccerAI software, in addition to having a large portion of code dedicated to perform various kind of robot or AI skills such as moving to a certain location, getting ball, passing ball to another robot, making a calculated goal shot, etc., on an abstract level mainly runs the state machine illustrated below. The core of the soccer-playing AI algorithm is constructing probability map for estimating the probability of successful pass, successful goal shot, and having an attack advantage, and then construct a graph connecting each robots, whose weight is the probability instead of locational distance, and using algorithm such as Dijkstra to find the optimal path for carrying out an attack plan.



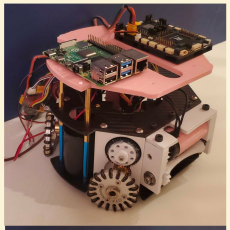
Hardware

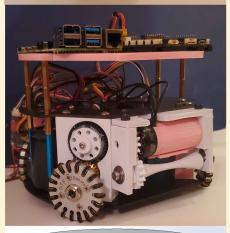
System Overview

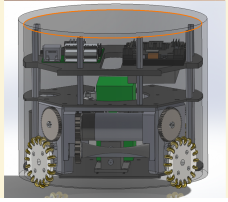


Mechnical

Our robot is made of carbon-fiber and 3D printed components to ensure low mass and a low cost of manufacturing. This robot has two main mechanisms for matchwinning: the kicker and the dribbler. The kicker is powered by a standard solenoid which can achieve the max rated speed of 6.5 meters per second. The dribbler is a 3D printed component that controls the ball while the robot is moving. The dribbler uses a XING-E Pro 2207 1800KV Brushless Motor, which provides high rotational speed since it would be typically used for quadcopter robots. Next, to ensure our robot is up to date with the current motion standards, we are going with a 3.2 gear ratio that ultimately moves the robot at 5 meters per second.







Electrical

Our electrical design for the kicker circuit is based on the LT3751 Flyback Converter topology. Pairing that up with a voltage regulator in the form of the LTC2955CTS8-1hot swap controller and the LTC4231CMS-2 and you got yourself a modern SSL boost up circuit. We are using a 2700uF mega capacitor to discharge high current into our standard solenoids. Our switching circuit uses a high power latch for the first powering stage and the IKB40N65ES5ATMA1 for the switching stages. These branches are, of course, protected by a series of high current rated diodes. Electrifying!

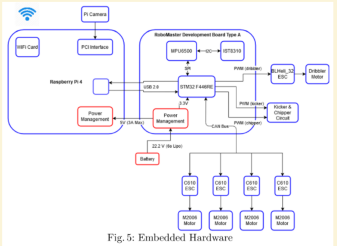
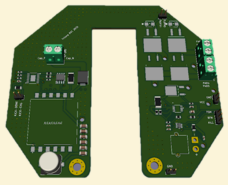


Fig. 5: Embedded Hardware



Team Tritons RCSC

Robot Component	Details
Embedded Computer	Broadcom BCM2711 Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz (Embedded in Raspberry Pi 4B)
Embedded Microcontroller	STM32F427IIH6 Cortex-M4 (ARM) 32-bit C @ 180 MHz (Embedded in DJI RoboMaster Development Board Type A [abbrev. as RM])
IMU System (9DOF)	MPU6500 6DOF IMU (Embedded in RM), IST8310 3DOF Magnetometer (Embedded in RM)
On-Robot Camera	8 Megapixel Pi Camera
Proximity Sensor	ST VL53L1X ToF (Not included in the current prototype, but will appear in a future upgrade to detect ball-holding status)
Communication	WiFi between standard home router and our PC
Main Motors	DJI M2006 Motor with built-in encoders, Max 500 rpm, Max 44W, 416rpm at 1 Nm, @24V
Gear Ratio	3.33, wheel speed up to 1385.28 rpm
ESCs	DJI C610 32-bit FOC ESC (interfaced with CAN BUS), @24V, @Max 10A
Wheels	GTF 50mm Omni Wheel
Dribbler Motor & ESC	T-MOTOR MT2212-13 980KV Brushless Motor (current prototype), XING-E 2207 1800KV Brushless Motor (future upgrade), ICQUANZX ESC BLHeli_S 6s 35A
Kicker Circuit	LT3751 Capacitor Charger Controller IC, GA3459-BL Flyback Transformer (turn ratio 1:10), IGBT switch (FZT755TA PNP + FDS2582 NMOS), 2700 Capacitor, @12v operating voltage, boost to 130V in 272 ms
Servo	WEISE DS3218 Servo @5V 20KG
Power Supply	22.2 V 6s LiPo, 1550 mAh, 100C

Table 1: Robot Specification Table