Walking:

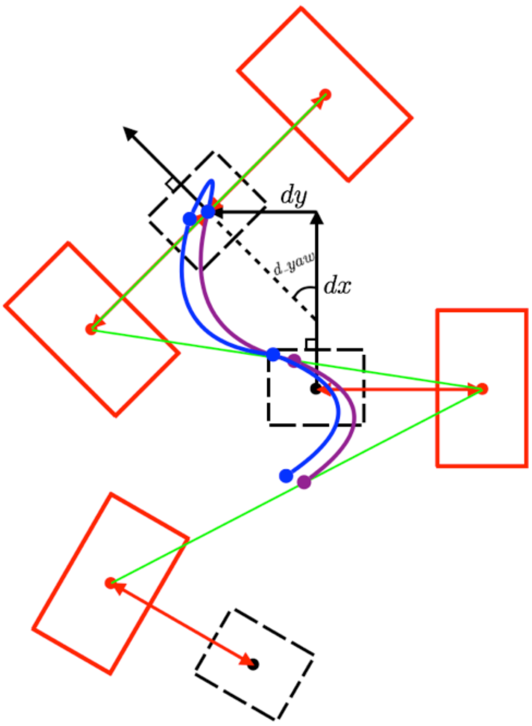
Our walking algorithm can be divided into four modules. The first module is kinematic calculation module. It can calculate the solution of inverse kinematic and the forward kinematic of legs and body. The starting point of kinematic solution is the body center point.

The second module is interpolation module. We used cubic interpolation which is different from cubic spline interpolation. We not only specify the starting and ending points of the abscissa and ordinate, but also specify the slope at the starting and ending points, which allows us to more actively control the speed of the steering gear at a certain point. The interpolation module is used to make the position of the steering gear change smoother.

The third module is our kinematic model of the whole body of the robot. We chose a robot state to analyze the kinematics of the robot. It is the state in which the robot stands on one leg. We built a three-mass model for the robot, assuming that the mass of the robot is concentrated on three mass points, the upper body center of mass, the left leg center of mass, and the right leg center of mass. In the static state, the input of this model is the position and pose of the robot's hanging foot and the position of the body's center of mass. Using the information of the fixed position of the leg center of mass, we can calculate the position of the upper body centroid according to the theory of the particle system, and then calculate the center of the body. During the process, we can also choose the pose of the upper body of the robot, which can be upright or fit the whole body, or as an input to the model. The position, and then the inverse kinematics module can be used to calculate the steering value of the whole body joint to complete the stable task of the robot in a static state.

Finally, the fourth module is the waling module. The walking algorithm of the robot is divided into two parts: gait pattern generation and walking stabilizer. The kinematic model of the robot is the basic of the gait pattern generation. We approximate the robot model as a linear inverted pendulum model, that is, keep the height of the robot's centroid constant. We use the dynamic equation of the linear inverted pendulum to solve the centroid trajectory in a gait unit, and then use the interpolation module to calculate the trajectory of hanging leg. The trajectory is input into the multi-link model of the robot, and the joint steering value of the robot can be calculated to achieve bipedal walking.

The figure below shows our walking algorithm. The black dotted box is the ideal displacement of the robot at the level of path planning and does not represent the entity. The red box is the foothold of the robot. It is located at a distance of 7cm on both sides of the black dotted box. The side is determined by which foot should be taken next. The thin green line is the line connecting two adjacent points. The purple point is the midpoint of the thin green line. As the starting point of the centroid planning for a single gait cycle, the purple curve is the centroid trajectory. The blue curve is used for acceleration and deceleration of the robot. The x and y offsets are added to the purple curve for acceleration and deceleration.



dy
dx
d_yaw