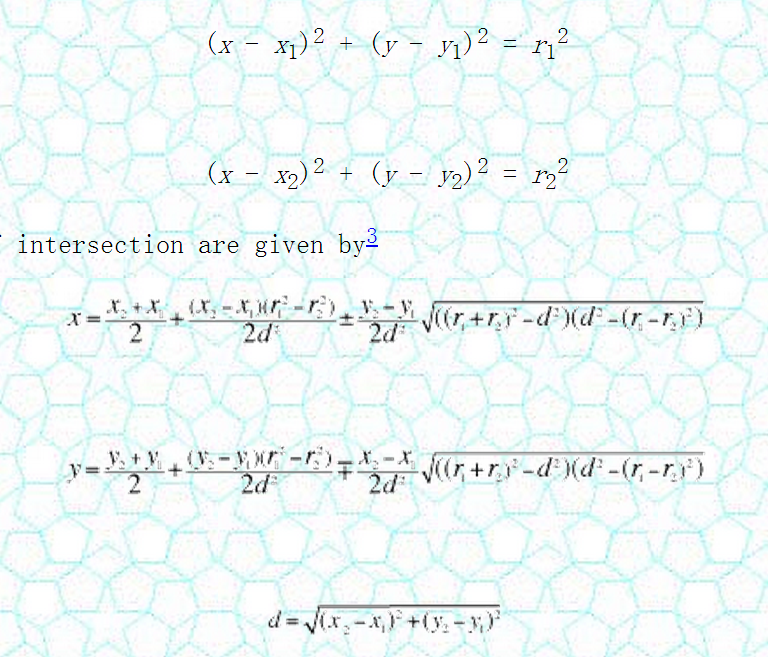
1. Circles Intersection



2. Forward Kinematics (Theta in degrees)

int delta\_calcForward(float theta1, float theta2, float theta3, float &x0, float &y0, float &z0) {

float t = (f-e)\*tan30/2;

float dtr = pi/(float)180.0;

theta1 \*= dtr;

theta2 \*= dtr;

theta3 \*= dtr;

float y1 = -(t + rf\*cos(theta1));

float z1 = -rf\*sin(theta1);

float y2 = (t + rf\*cos(theta2))\*sin30;

float x2 = y2\*tan60;

float z2 = -rf\*sin(theta2);

float y3 = (t + rf\*cos(theta3))\*sin30;

float x3 = -y3\*tan60;

float z3 = -rf\*sin(theta3);

float dnm = (y2-y1)\*x3-(y3-y1)\*x2;

float w1 = y1\*y1 + z1\*z1;

float w2 = x2\*x2 + y2\*y2 + z2\*z2;

float w3 = x3\*x3 + y3\*y3 + z3\*z3;

// x = (a1\*z + b1)/dnm

float a1 = (z2-z1)\*(y3-y1)-(z3-z1)\*(y2-y1);

float b1 = -((w2-w1)\*(y3-y1)-(w3-w1)\*(y2-y1))/2.0;

// y = (a2\*z + b2)/dnm;

float a2 = -(z2-z1)\*x3+(z3-z1)\*x2;

float b2 = ((w2-w1)\*x3 - (w3-w1)\*x2)/2.0;

// a\*z^2 + b\*z + c = 0

float a = a1\*a1 + a2\*a2 + dnm\*dnm;

float b = 2\*(a1\*b1 + a2\*(b2-y1\*dnm) - z1\*dnm\*dnm);

float c = (b2-y1\*dnm)\*(b2-y1\*dnm) + b1\*b1 + dnm\*dnm\*(z1\*z1 - re\*re);

// discriminant

float d = b\*b - (float)4.0\*a\*c;

if (d < 0) return -1; // non-existing point

z0 = -(float)0.5\*(b+sqrt(d))/a;

x0 = (a1\*z0 + b1)/dnm;

y0 = (a2\*z0 + b2)/dnm;

return 0;

}