**Cálculo do ângulo em relação à linha e da distância à linha**



Ângulo com a reta:

tg(a) = (S1 – S2)/D12

No código corresponde à linha:

alfa = atan((robot.IRLine\_Front.dist\_center - robot.IRLine\_Back.dist\_center)/D12);

Distância à reta (nesta figura D2z é negativo):

Equação da reta que passa por (D12 + D2z, S1), a tracejado: y – S1 = tg(a).(x – (D12 + D2z) )

No ponto (0, Dy) teremos: Dy – S1 = tg(a).(– (D12 + D2z)) ⬄ Dy = tg(a).(–– (D12 + D2z)) + S1

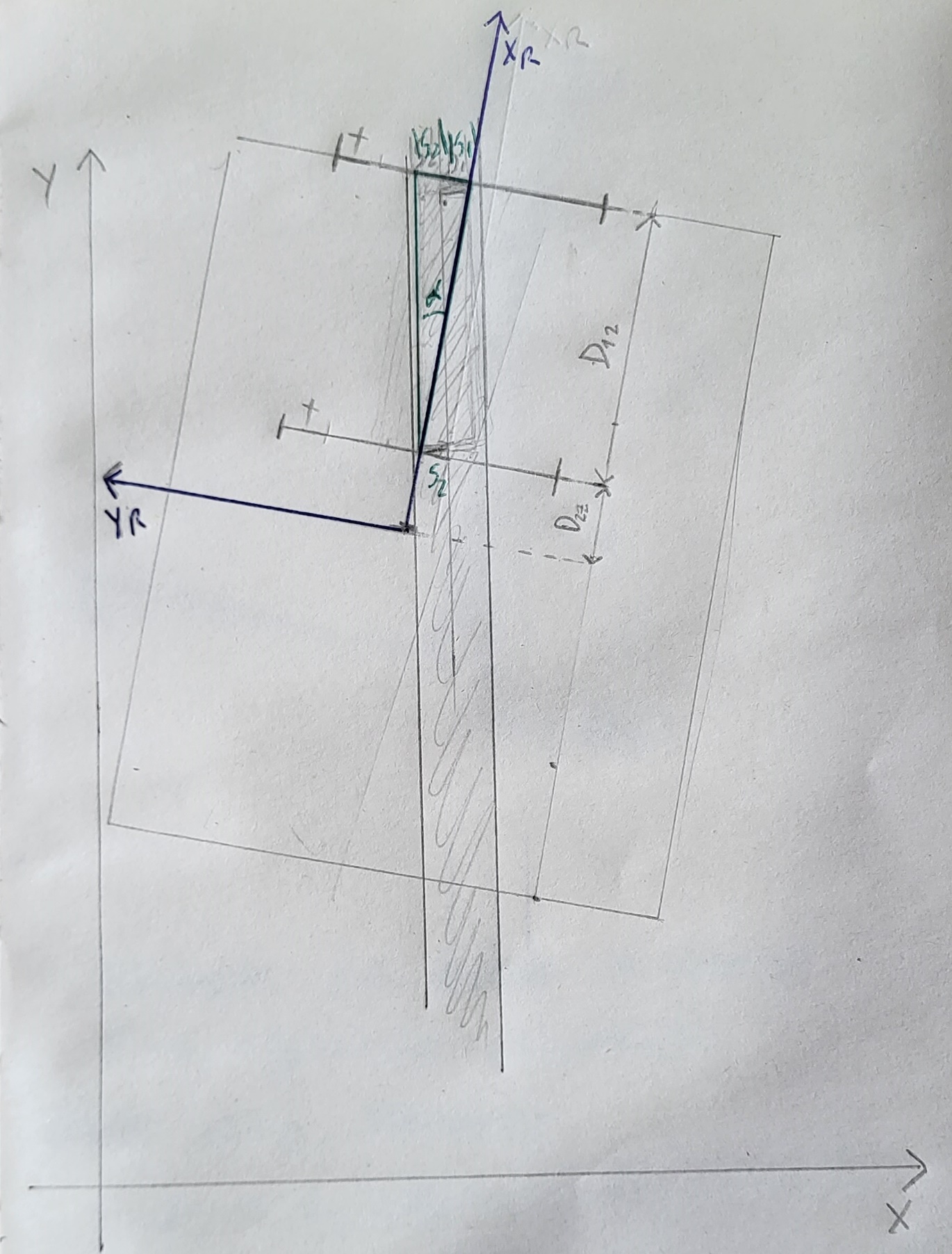
A distância, com sinal, será: dc = Dy.cos(a)

Cálculo do ângulo que o robô faz com a reta (triangulo verde).

D12 – distância enre os dois sensores

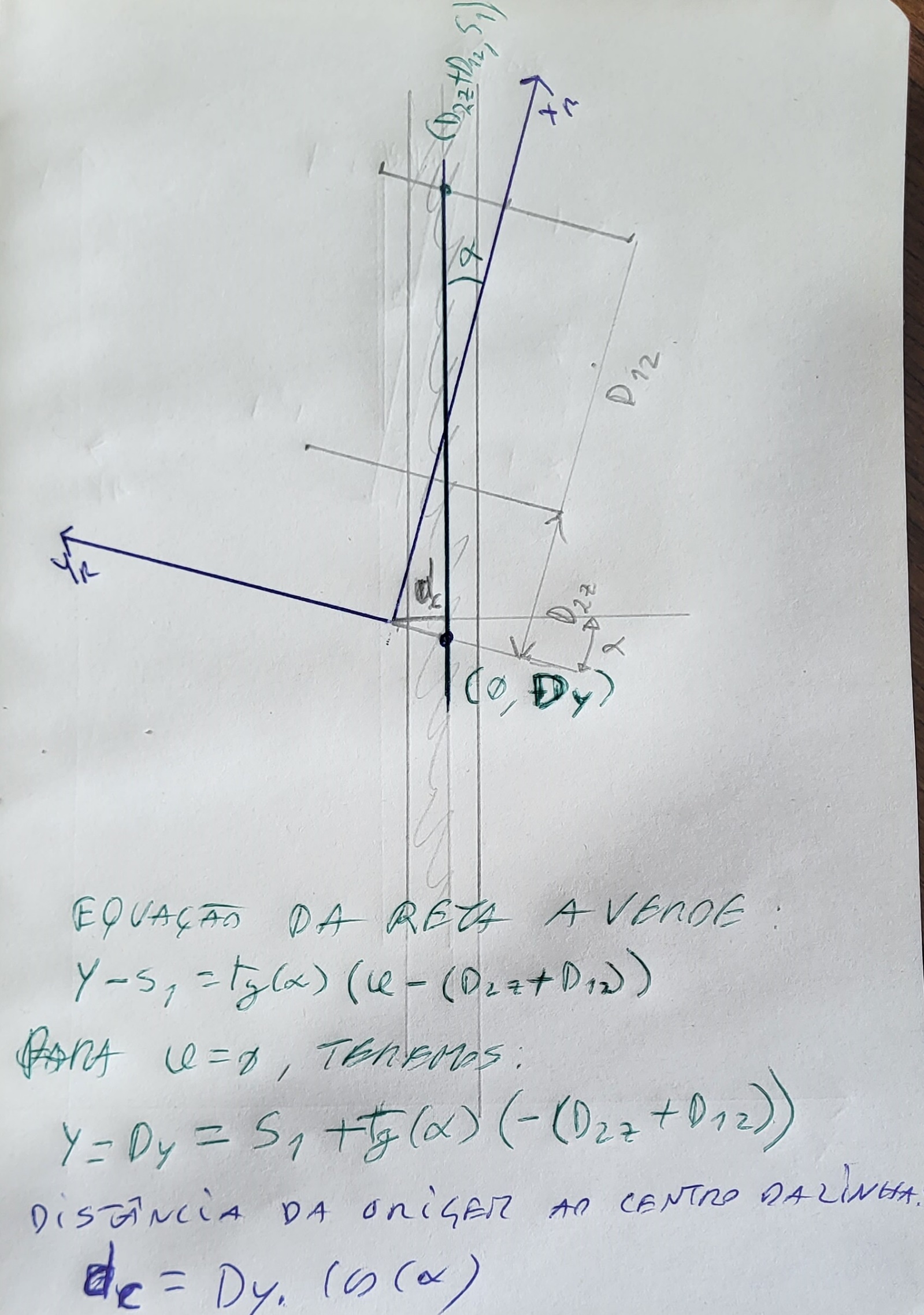
D2z – coordenada X do sensor 2 à origem do referencial do robô.

Angulo coma reta:



Uma imagem com desenho, esboço, texto, desenhos de criança

Os conteúdos gerados por IA poderão estar incorretos.

****

**Ficheiro « robot.cpp » :**

void robot\_t::localization(void)

{

float alfa;

// Line Crosses must be first to calculate variable "total" and "last\_total"

// transições detetadas pelo sensor da frente e respetivas correções da estimativa quando o robô está a percorrer a reta vertical com x=-0,724

if (robot.IRLine\_Front.SenseIRLineCrosses() == true) {

  if  ((abs(xe -(-0.724)) < TOL\_DIST\_LOC) && (abs(thetae - PI/2) < TOL\_TH\_LOC/3)) {

    // CAUTION! difference between marks must be greater than TOL\_DIST\_LOC

    if (abs(ye -(-0.265)) < TOL\_DIST\_LOC) ye = -0.265;

    if (abs(ye -(-0.237)) < TOL\_DIST\_LOC) ye = -0.237;

    if (abs(ye -(-0.116)) < TOL\_DIST\_LOC) ye = -0.116;

    if (abs(ye -(-0.083)) < TOL\_DIST\_LOC) ye = -0.083;

    if (abs(ye -0.255) < TOL\_DIST\_LOC) ye = 0.255;

    if (abs(ye -0.285) < TOL\_DIST\_LOC) ye = 0.285;

  }

}

// transições detetadas pelo segundo sensor e respetivas correções da estimativa quando o robô está a percorrer a reta vertical com x=-0,724

if (robot.IRLine\_Back.SenseIRLineCrosses() == true) {

  if  ((abs(xe -(-0.724)) < TOL\_DIST\_LOC) && (abs(thetae - PI/2) < TOL\_TH\_LOC/3)){

    // CAUTION! difference between marks must be greater than TOL\_DIST\_LOC

    if (abs(ye -(-0.308)) < TOL\_DIST\_LOC) ye = -0.308;

    if (abs(ye -(-0.279)) < TOL\_DIST\_LOC) ye = -0.279;

    if (abs(ye -(-0.083)) < TOL\_DIST\_LOC) ye = -0.083;

    if (abs(ye -(-0.057)) < TOL\_DIST\_LOC) ye = -0.057;

    if (abs(ye -0.063) < TOL\_DIST\_LOC) ye = 0.063;

    if (abs(ye -0.096) < TOL\_DIST\_LOC) ye = 0.096;

  }

}

// cálculo da distância ao centro da linha dada pelo sensor da frente

robot.IRLine\_Back.dist\_center = 999; // 999 - no line detection

if (robot.IRLine\_Back.total < robot.IRLine\_Back.cross\_total\_tresh ) {// near crosses! Invalid Edges!

  if (robot.IRLine\_Back.calcIRLineEdgeLeft() == true) {

    robot.IRLine\_Back.dist\_center = robot.IRLine\_Back.pos\_left;

    if (robot.IRLine\_Back.calcIRLineEdgeRight() == true) robot.IRLine\_Back.dist\_center = (robot.IRLine\_Back.pos\_left + robot.IRLine\_Back.pos\_right)/2;

  } else {

    if (robot.IRLine\_Back.calcIRLineEdgeRight() == true) robot.IRLine\_Back.dist\_center = robot.IRLine\_Back.pos\_right;

  }

}

// cálculo da distância ao centro da linha dada pelo segundo sensor

robot.IRLine\_Front.dist\_center = 999; // 999 - no line detection

if (robot.IRLine\_Front.total < robot.IRLine\_Front.cross\_total\_tresh ) {// near crosses! Invalid Edges!

  if (robot.IRLine\_Front.calcIRLineEdgeLeft() == true) {

    robot.IRLine\_Front.dist\_center = robot.IRLine\_Front.pos\_left;

    if (robot.IRLine\_Front.calcIRLineEdgeRight() == true) robot.IRLine\_Front.dist\_center = (robot.IRLine\_Front.pos\_left + robot.IRLine\_Front.pos\_right)/2;

  } else {

    if (robot.IRLine\_Front.calcIRLineEdgeRight() == true) robot.IRLine\_Front.dist\_center = robot.IRLine\_Front.pos\_right;

  }

}

// caso sejam detetadas transições nos dosi sensores, calcula o ângulo e distância (com sinal) ao centro da linha

if ((robot.IRLine\_Front.dist\_center != 999) && (robot.IRLine\_Back.dist\_center != 999)) {  // detection in both sensors

  alfa = atan((robot.IRLine\_Front.dist\_center - robot.IRLine\_Back.dist\_center)/D12);

  if (abs(thetae - PI/2) < TOL\_TH\_LOC){

    if (abs(xe -(-0.724)) < TOL\_DIST\_LOC) {

        xe = -0.724 + (robot.IRLine\_Front.dist\_center + tan(alfa)\*(-(D12 + D2z)))\*cos(alfa);

        thetae = PI/2 - alfa;

    }

  }

}

}