Settings

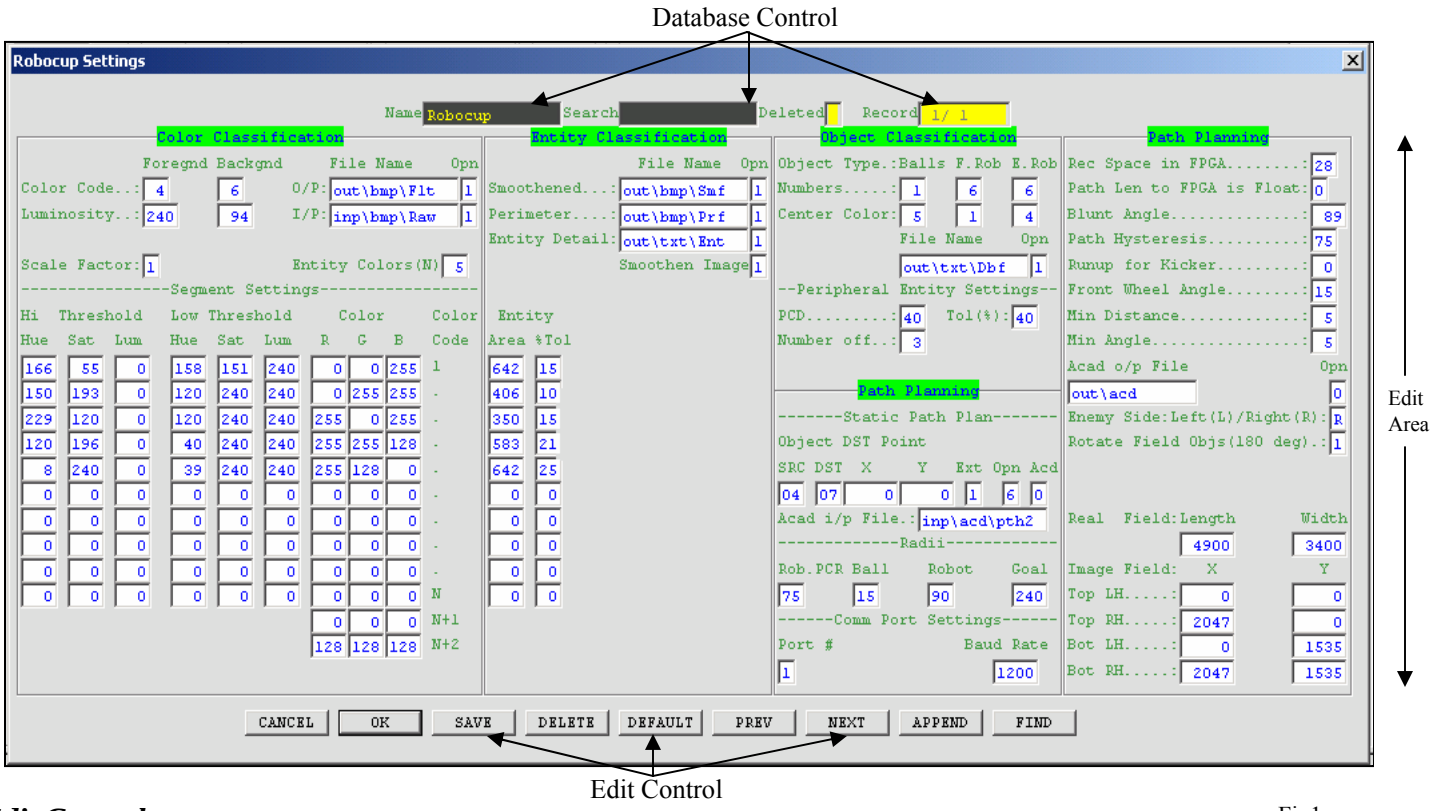

Fig1

Edit Control

- **CANCEL**
  Discard edits. Exit, with a return value of '0' and without a validity check.
- **OK**
  Perform a validity check. If all entries are valid, save the record and exit editing with a return value of '1', if the record is not deleted and '0' if it is. If invalid entries are found, they are colored red and editing continued.
- **SAVE**
  Perform a validity check and continue editing . If all entries are valid, save record. Invalid entries are colored red.
- **DELETE**
  Delete/Recall a record and continue editing. Pressing this button when the 'Deleted' box is ' ', will delete the record and save it w/o a validity check. When the 'Deleted' box is '*', after a successful validity check, a deleted record is recalled and saved. deleted records are removed from the database only after exiting by the 'Ok' or 'Cancel' options.
- **DEFAULT**
  Insert default data in all the edit boxes(except the Key field box) and continue editing.
- **PREV**
  Skip to previous record, if one is present and the Edit boxes of the current record have not been changed, display it on the screen and continue editing.
- **NEXT**
  Skip to next record, if Edit boxes of current record have not changed, display on screen and continue editing. If end of file is encountered an empty screen for a new record is displayed
- **APPEND**
  Append new record, if Edit boxes of current record have not changed, display on screen and continue editing.
- **FIND**
  Search for a record specified in the 'Search' Edit box, if Edit boxes of current record have not changed, display on screen and continue editing. If the end of file is encountered an empty screen for a new record is displayed.

Note: Only if the return code is '1' should the entries be used for further processing.

Database Control

- **Name**
  A 14 character wide, database Key Field with a non-numeric in the 1st position
- **Search**
  A 14 character wide, Search string for the 'Key Field'. May be a numeric or character entry. A numeric entry specifies the record # to edit. A string entry, causes the file to be searched, for the first record with a matching 'Key Field'.
- **Deleted**
  A one character wide entry. '*' in this box indicates that the record on display is deleted.
- **Record**
  Displays the current record #, on the left hand side of '/' and total records, on the right hand side. When an empty screen for a new record is displayed, 'NEW' is suffixed.

Color Classification Settings

Objects referred to as ‘member’, are members of the ‘FLTR’ structure, passed as an i/p to GetColor().
The File Names mentioned here have a width of 13. See the 'Color Classification' document for details.

- **Foregnd**
  3 digit integer values for ‘Foreground’ settings:-.
  - Color Code-Color code. Range:1 to N+2, taken into the ’eclr[1]’ member.
  - Luminosity-Luminosity limit. Range:0 to 240, taken into the ’eint[1]’ member.
- **Backgnd**
  3 digit integer values for the 'Background' settings:-.
  - Color Code-Color code. Range:1 to N+2, taken into the ’eclr[0]’ member.
  - Luminosity-Luminosity limit. Range:0 to 240, taken into the ’eint[0]’ member.
- **File Name and Opn**
  - I/P
    Opn and File name are taken into the ‘fili’ and ‘rfile’ members *resp.* Using the ‘SnapShot’ main menu option, an image is saved into the file specified and used when the ‘Acd’ box=0.
  - O/P
    Opn and File name are taken into the ‘fprn’ and ‘fnam’ members *resp*.
- **Scale Factor**-Taken into the 'scale' member.
- **Entity Colors(N)**
  Defines the effective depth of Table 1, below which Table 2 is appended. N has range:1 <= N <= 10 and is taken into the ‘ncc’ member.

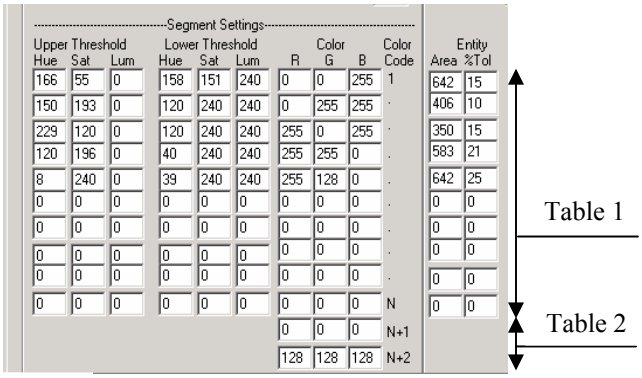

Fig2

- **Segment Settings**
  - Upper and Lower Threshold
    Each row in Table 1(fig 2) represents one of 'N' color segments defined for the raw image. The values in these boxes(integer between 0 and 240) are the upper and lower threshold values of Hue, Saturation and Luminosity for the segment. Taken into HUE, SAT and LUM members.
  - Color
    Each row in Table 1(fig 2) contains the Red, Green and Blue color components(integer between 0 and 255) of one of 'N' color segments defined for the raw image. Table 2 contains additional colors whose color codes may be used for the 'Backgnd' and 'Foregnd' settings. Taken into the 'color'
    sub-member of the 'clsc[]' member.

Entity Classification Settings

Objects referred to as ‘member’, are members of the 'ENTT' structure, passed as an i/p to GetEntity().
The File Names mentioned here have a width of 13. See the 'Entity Classification' document for details.

- **File Name and Opn**
  - Smoothened -Opn and File name are taken into the 'sprn' and 'sfile' members *resp*.
  - Perimeter -Opn and File name are taken into the 'pprn' and 'pfile' members *resp*.
  - Entity Details -Opn and File name are taken into the 'eprn' and 'efile' members *resp*.
- **Smoothen Image**-Taken into the 'smth' member
- **Entity Area and Tolerance**
  Each row in Table 1(fig 2) represents one of 'N' color segments and entity types defined for the raw image. The values in these boxes are the Area(in pixels, an integer between 0 and 999) and Area Tolerance(%, an integer between 0 and 50) values of each entity type. These are converted into upper and lower area thresholds and taken into the 'pcnt[]' member

Object Classification Settings

Objects referred to as ‘member’, are members of the 'ROBB' structure, passed as an i/p to GetObject().
The File Names mentioned here have a width of 13. See the 'Object Classification' document for details.

- **Object Numbers and Center Color**
  ‘Object Numbers’ is the number of objects, of each type, expected to be found in the image. Taken into the ‘nob’ member and used for error detection only. ‘Center Color’ is the color code(1 to ncc) of the center entity of each object type(taken into the 'cc[]' member).
- **File Name and Opn**-Opn and File name are taken into the 'dprn' and 'dfile' members *resp*.
- **Peripheral Entity Settings**
  - PCD & Tol(%)
    PCD is the mean radial distance between the cg of a center entity of an object and the cg of a peripheral entity, for the peripheral entity to be considered belonging to the object. Tol(%) is a bi-polar tolerance used to calculate the upper and lower limits of the mean radial distance. The upper and lower limits are taken into the 'upcd' and 'lpcd' members *resp.*
  - Number off
    Number of Peripheral Entities. Used to determine Peripheral Entity colors, taken into the ‘fc[0][]’ member.

Path Plan Settings

Objects referred to as ‘member’, are members of the 'PSET' structure, passed as an i/p to GetEntity().
The File Names mentioned here have a width of 13.

'Object Classification' generates a Field Object list containing the location of robots & ball & orientation point of friendly robots. Field objects are given numbers. Friendly robot numbering depends on the 'Acd' option and others are given sequential numbers thereafter. After 'Object Classification and before Path Planning, the Enemy goal is added and all the objects are given a radius. The Field and its objects are skew corrected and rotated depending on 'Acd'

Path Types

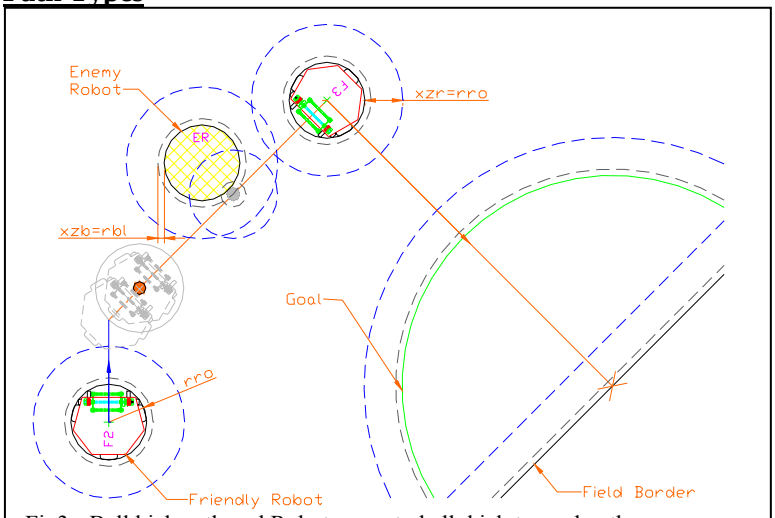

Fig3 - Ball kick path and Robot move to ball, kick to goal path

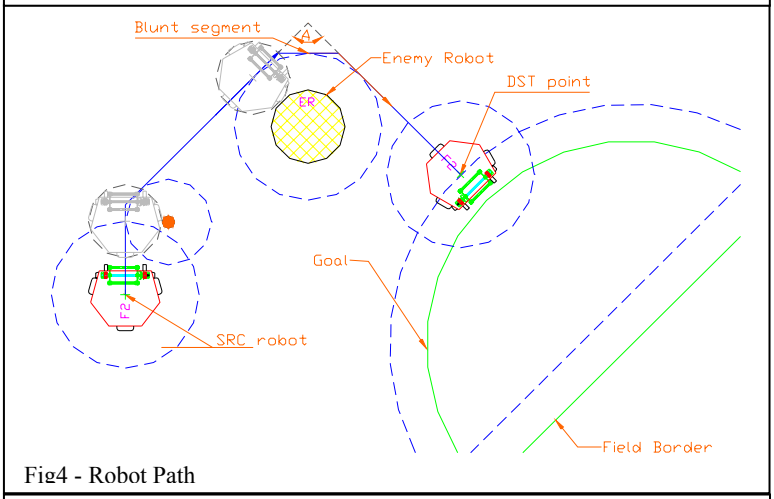

Fig4 - Robot Path

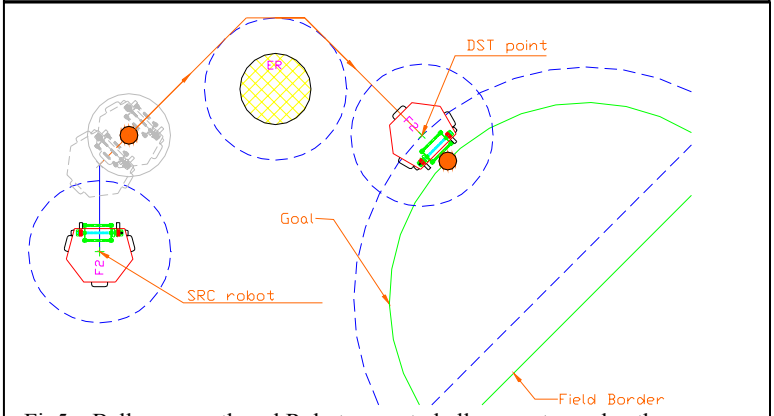

Fig5 – Ball move path and Robot move to ball, move to goal path

A Cumulative Path is a series of end to end connected Path(line) segments from the center of the source object to a destination point. Paths can be planned on a static or dynamic field. The former requires the 'Frame Processing' menu option, the latter requires the 'Game Processing' menu option. The i/p for both are specified by the 'Acd' option below.

There are basic and composite paths.

Basic paths

- **Ball kick path** to goal is shown in red(fig 3).
- **Ball move path** to goal is shown in red(fig 5).
- **Robot path** of robot f2, shown in blue to ball(fig 3,5), to DST point(fig 4).

Composite paths

- **Robot move to ball, kick to goal path** - fig 3
  A friendly robot is selected as the source & its composite path('Ball kick path' to goal and robot path to ball) is made by connecting the above two paths with a connecting path(red dotted). The process is repeated for all friendly robots & the one with smallest composite path selected. The dotted red line connecting the two basic paths is the **kicker on segment** where the robot turns on the kicknd moves towards the ball.
- **Robot move to ball, move to goal path** - fig 5
  A friendly robot is selected as the source & its composite path('Ball move, path' to goal and robot path to ball) is made by connecting the above two paths with a connecting path(red dotted). The process is repeated for all friendly robots & the one with smallest composite path selected.

- SRC - Field Object #. Normally friendly robot, but may be enemy robot or ball. Taken into ‘srco’ member.
- DST - Field Object # at whose center a Cumulative Path terminates. Taken into ‘dsto’ member

- DST Point - Co-ordinate of point at which a Cumulative Path terminates. Taken into ‘dstp’ member.
- Ext
  When 'Ext'=1 the Cumulative Path terminates exactly as specified and when 0 termination may be near the point specified. Taken into ‘exct’ member.
- Opn
  Static Path Plan option. Taken into ‘cst’ member:-
  0 - Move robots on **custom Paths**, specified in 'Acad i/p File'. SRC, DST, ‘DST point’ and Exact are unused.
  1 - **Ball kick path** to the goal. Kicker robot is SRC. DST and ‘DST point’ are unused, no RF
  2 - **Ball move path** to goal. Robot is SRC. DST and ‘DST point’ are unused, no RF.
  3 - **Robot path** to ‘DST point’. Robot is SRC. DST is unused
  4 - **Robot path** to ‘DST’. Robot is SRC. ‘DST point’ is unused.
  5 - **Robot move to ball, kick to goal**. SRC, DST, ‘DST point’ and Exact are unused.
  6 - **Robot move to ball, move to goal**. SRC, DST, ‘DST point’ and Exact are unused.
  7 - At the location specified in the 'Acad o/p File' edit box, generates an autocad script file(fld0.scr) for the skew corrected & rotated field & objects, superimposed on the original field(in grey) w/o objects. This option ignores the 'Opn' box of the 'Acad o/p File' box.

- Acd – Use 'Acad i/p file' ?. Taken into ‘acd’ member. May be 0 or 1
  1 - Yes. The first operation before Static path planning is 'Object Classification' whose input is taken from the 'Acad i/p File'. Friendly robots numbers correspond to the object name e.g friendly robot 'f1' is given 1.
  0 - No. The first operation before Static or Dynamic path planning is 'Color Classification' whose input depends on the 'Opn' option of the 'I/P File Name' box in 'Color Classification Settings', when 1 it is taken from the image file specified & when 0, from the camera. Friendly robot numbers correspond to the color code on their surface. The field & objects are skew corrected and rotated(See 'Rotate Field Objects').

- Acad i/p File
  When 'Acd'=1, the autocad i/p for 'Object Classification' is taken from a file whose name and location is mentioned in this edit box and taken into the ‘afile’ member. It is created as follows(see fig6):-
  1>Create a new autocad drawing.
  2>Using the 'XATTACH' command insert the field(FIELD), friendly robots(F1 to F6), enemy robots(ER), ball(BALL) and enemy goal(GOAL), external drawing references(XREFs) into it.
  3>Rotate the friendly robots as required and make a cumulative path for one or more friendly robots with end to end connected path segments of the following colors(consecutive segments may have the same color):-
    - Green - Forward motion
    - Red - Reverse motion
    - Blue - Forward motion with kicker on

  4>Closed paths(end point of a path segment is coincident on the start point of another path segment) can be made, however the segment taken by the robot at the point of coincidence is indeterminate. If the segment selected in a closed path is important, displace the end of the last segment by 0.0001mm(approx) from the start of the first.
  5>Use the 'DBLIST' command and copy-paste its output into an MS-WORD document saved as an MS-DOS text file. Enter the name and location of this file into the 'Acad i/p File' edit box.

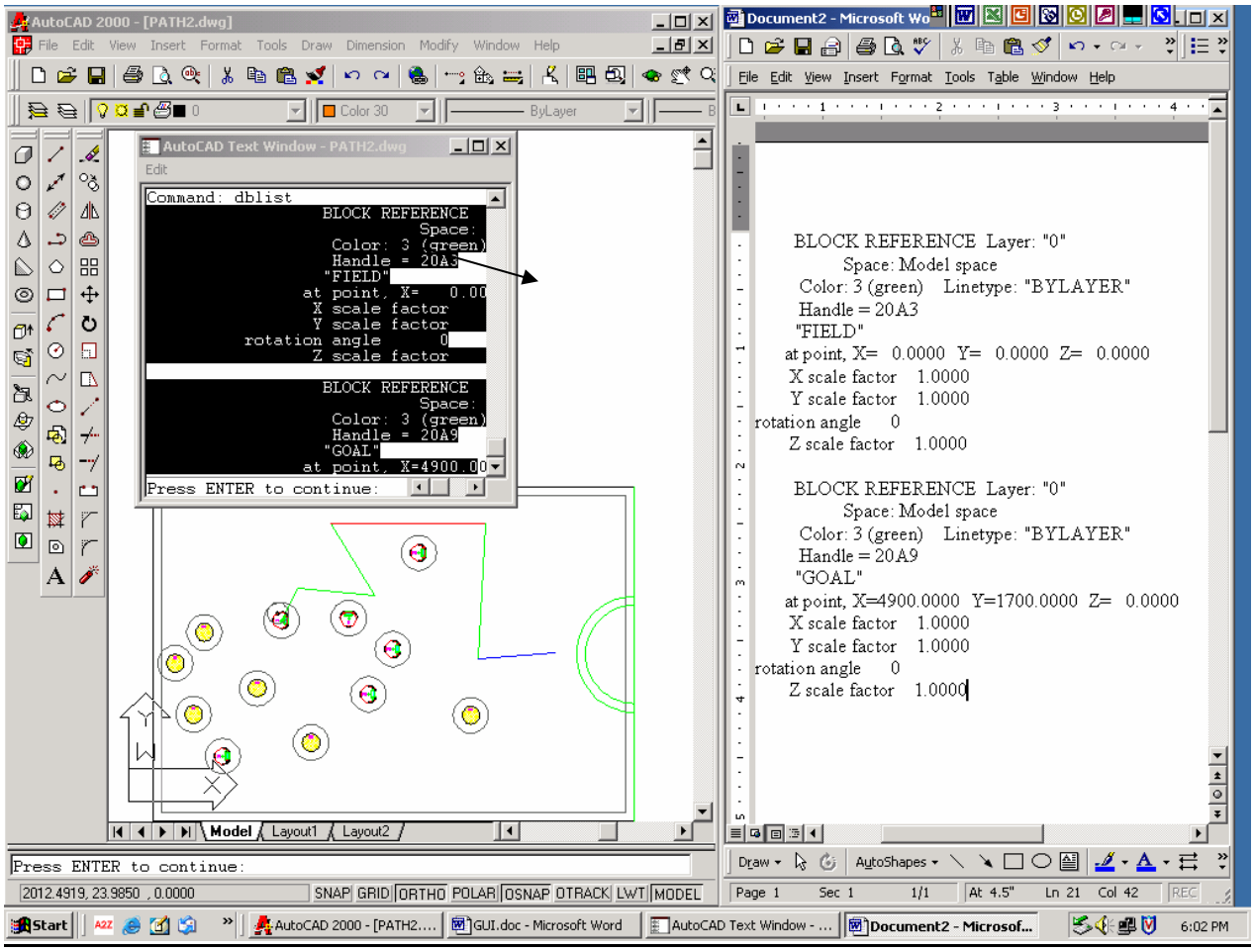

Fig6 – Autocad i/p file generation

- **<u>Radii</u>**
  - <u>Goal</u> - Radius of Goal in 'mm'. Taken into 'gr' member
  - <u>Ball</u>
    Fig 3 shows a 'Ball kick path'. After making the source robot(kicker) transparent, an exclusion zone(xzb) equal to the **Ball radius**(rbl) is constructed around field objects, to prevent collisions with the ball, by ensuring that path segments do not intersect the exclusion zone(xzb). If collision is observed, 'rbl'(in mm) may be increased. 'rbl' is taken into the 'rbl' member.
  - <u>Robot</u>
    Fig 4 shows a 'Robot path' moving the source(SRC) robot 'f3' to destination(DST) point. An exclusion zone(xzr) equal to the **Robot radius**(rro) is constructed around field objects, to prevent collisions with the SRC object, by ensuring that path segments do not intersect the exclusion zone(xzr). If collision is observed, 'rro'(in mm) may be increased. 'rro' is taken into the 'rro' member.

- Rob PCR
  When a robot rotates, the inner and outer rim of its wheels slip, with the outer rim rotating faster than the inner. The plane passing through the wheel, between the inner and outer rim, doesn't slip and is used to calculate the circumferential displacement for a given angular displacement. A circle tangent to this plane with its centre at the robot center, is the pitch circle and its radius the **P**itch **C**ircle **R**adius(**PCR)**. PCR in 'mm' is taken into the 'fc2' member.

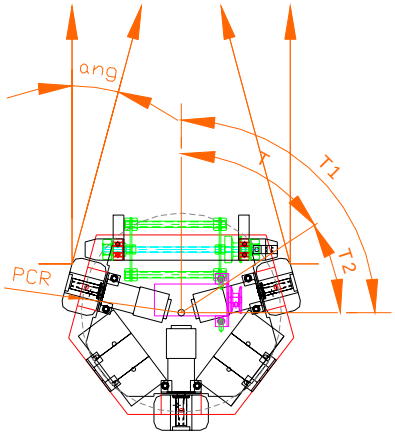

Fig7a Robot at position 1

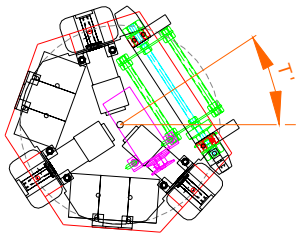

Fig7b Robot at position 2

The Circumferential displacement '**D**' required to rotate the robot from position 1 to position 2 is:-
C=2π*PCR -- Pitch Circle Circumference
D=2π*PCR*(T/2π)
D=PCR*T -- See doc on MAKE(), generation of serial Tx file

Where 'T' is calculated as follows:-
Tan(T)=Tan(T2-T1)=[Tan(T2)-Tan(T1)]/[1+Tan(T2)*Tan(T1)]
Let:-
Tan(T2)=m2 , Tan(T1)=m1 , Tan(T)=m
Then:-
m=[m2-m1]/[1+m2*m1]
and
$T=Tan^{-1}(m)$

- **Com Port Settings**
  - Port Number - Serial port#: 1 for COM1. "COM1" string taken into 'comp' member
  - Baud Rate - Serial port baud rate. Taken into 'baud' member.
- **General Settings**
  - Rec Space in FPGA
    A record of data sent by RF to the robot processor(FPGA) is a control word and a path length.
    The number of such records that can be sent to the FPGA depends on how the FPGA has been configured and is currently set to (x=28). The Game Processing and Static Path Plan menu options transmit 'x' and 1 records *resp.*
    This setting must be set to 'x' and is taken into the 'lct' member.

- Path Len to FPGA is Float
  The path length constituent of a record to data sent to the robot FPGA may be a 16 bit integer or a 32 bit float value(real4 format), depending on the configuration of the FPGA. The current setting must be(0) for integer and is taken into the 'real' member.
- Blunt Angle
  If the angle(A), between path segments of a **robot path** or **ball move path**, is less than this setting, the path is blunted as shown in Fig 4.The process of blunting continues recursively until the angles between the segments satisfy the constraint. Cos(Blunt Angle/2) is taken into the 'bfc' member.
- Path Hysteresis
  Below are two 'basic' paths for a robot to a destination. The shorter path has been determined, the longer path is being determined. Segments of this path(grey) are ignored once its length exceeds the shorter path by a percentage specified in the 'Path Hysteresis' edit box. Also applies to 'composite' paths, but not to the 'basic' paths within them. Path Hysteresis/100 is taken into the 'acc' member.

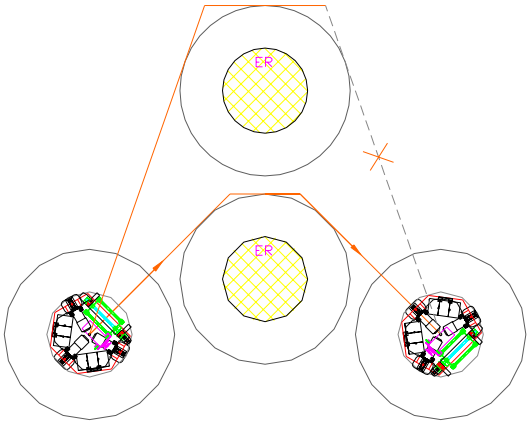

Fig9 - Path Hysteresis

- Runup for Kicker
  There are two ways of kicking & moving a ball. a> can be used only when the robot has a ball gripper.
  a>Robot rotates, the **ball moves** with it, to the direction required.
  b>**Ball is stationary**. The robot moves to position itself such that the ball is in the direction required (fig 3,5). If the robot has a rotatary kicker, the kicker takes time to accelerate and may require the kicker on segment(fig 3) to be longer, the excess length is entered in this box and taken into the 'otx' member.
- Front Wheel Angle
  When a robot moves in a straight line(translates), the angle of the front wheels('ang' in degrees), with the vertical, causes them to slip. The forward displacement(v1) of the robot is thus a component of the circumferential displacement of the forward wheels(v2) and is given as(see Fig 7):-
  The factor 0.9659258(below) is taken into the 'fc0' member.
  v1=v2*Cos(ang)
  When fc0=15 degrees:-
  v1=v2*0.9659258
- Min Distance
  If robot translation(mm) is less than this, translation doesn't occur. Taken into the 'fc1' member.
- Min Angle
  If robot rotation(degrees) is less than this, rotation doesn't occur. Taken into the 'fc3' mamber.

- Acad o/p File & Opn
  Used when the 'Static Path Plan Opn' =0 and 'Game Processing' menu option is selected or 'Static Path Plan Opn' =0 ....
  Script files for Autocad may be generated for the field and field objects(ball, friendly & enemy robots, goal). Overlayed on this are Tangents and Paths depending on the entry(0 to 3) in the 'Opn' edit box, taken into the 'sfo' member. The 'Acad o/p File' edit box requires a path for the file and is taken into the 'sfile' member. File names are generated internally and must not be included in the entry:-
  Opn
  0 - No file generated
  1 - Generate a file for all Tangents and a file for each Cumulative path
  2 - Generate a file for all Tangents and a file for all Cumulative paths
  3 - Generate a file for all Tangents, a file for each Cumulative path and a file for all Cumulative paths
  File naming convention is as follows:-
  - Tangent File Name
    1>"TanRr**S**" -Robot path (fig 4)
    2>"TanRb**S**" -Ball move path(fig 5)
    3>"TanB**S**" -Ball kick path (fig 3)
    Where **S** is: File number(S>=0)
  - Path File Name
    1>"PthRr**N_S**" -Robot path (fig 4)
    2>"PthRb**N_S**"-Ball move path(5)
    3>"PthB**N_S**" -Ball kick path (fig 3)
    Where **N** is: Kicker robot #(file type 2> and 3>), Robot #(file type 1>) and
    S is: File number(starts from 0 for every value of **N**).
- **Enemy Side**
  Enter 'L'(Left Hand Side) or 'R'(Right Hand Side). Taken into the 'esd' member.
- **Rotate Field Objects**
  When 1 and the 'Acd' box is 0, the field & its objects are rotated by 180˚. Unused when 'Acd'=1.
- **Real Field Dimensions**
  Dimensions of the real world field in 'mm'. Taken into the 'fxy' member.
- **Image Field corners**
  Spatial(see 'GetColor.doc') co-ordinates, in 'pixels', of the playing field corners in the image. Top left, top right, bottom left and bottom right, taken into the 'TL','TR','BL','BR' members *resp*.