Robot Behaviour Programmierung in URBI – Tipps und Tricks

V.V. Hafner, 05/2006
Release 0.9.8

This new release of URBI which is now compatible with ERS7-M3 robots includes two new main components: The new 'walk' device than enables a new walk coming from the robocup!

Here is a short description of the 'walk' device:

position control:
- walk.go(distance) : walk forward or backward meters
- walk.turn(angle) : turn degrees
- walk.side(distance) : sidestep meters
- walk.goto(y,x, theta) : combine the three previous functions

These functions block until move is terminated or interrupted by a new move command.
Neues walk device

speed control:
walk.speedforward(), walk.speedbackward(),
walk.speedleft(), walk.speedright(),
walk.speedsideleft(), walk.speedsideright()
 will move in the given direction at speed defined in walk.speed (between 0 and 1)

walk.speed(speed), walk.speedturn(speed) and walk.speedside(speed) will make the corresponding move, but at the signed speed passed as their parameter.

These functions can be mixed: the orders are averaged with each other.
These functions never stop: use tags, or stopif

odometry: the variables walk.x walk.y and walk.theta contains the position of the robot relative
to its start position. The variables can be reset to 0 by calling walk.reset()
Walk

Simple walk:
Using an original walk from the robocup, we extracted the two main Fourier coefficients for each joints. With blend = add mode, these two components are added on each joint to reproduce the original periodic oscillation:

def robot.walk(duration)
{
    echo "go for " + string(duration) + " secs";
    direction = 1;
    if (duration <0) {
        duration = - duration,
        direction = -1
    };

    walk: timeout(duration) {
        for &(x=1;x<=2;x++)
            for &(y=1;y<=2;y++)
                for &(j=1;j<=3;j++)
                    for &(d=1;d<=2;d++)
                        robot.leg[x][y][j] = walk.mean[x][j]
                            sin:walk.speed*walk.coef[d]
                            ampli:walk.amp[x][j][d]*4
                            phase:direction*walk.phase[x][y][j][d]+pi*(direction-1)/2
    }
};
Fourier Reihen

\[ f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(\omega_n t) + b_n \sin(\omega_n t)], \]

das gleiche in Exponential Schreibweise:

\[ f(t) = \sum_{n=-\infty}^{+\infty} c_n e^{i\omega_n t} \]