Fuzzy Path Tracking and Collision Avoidance for Automated Guided Vehicle for Matlab 5

% Fuzzy Path Tracking for Automated Guided Vehicle
% By Szilveszter Kovács
% e-mail: szkszilv@gold.uni-miskolc.hu

Add the directory of AGVFA to the Matlab Path and

Run RUNAGV.m in the Matlab Command Window to start.
(The basic Matlab environment is sufficient, no special toolboxes are required)

Brief usage:

You can start or stop the simulation by pressing the "Start" or "Stop" buttons. The simulation is stopping automatically in case of reaching the end of the path, or in case of collision (the actual situation is indicated under the "Start" button). In case of the simulation is stopped, you can edit the path, and the AGV parameters of the AGV by turning to Editing mode (pressing the "Edit" button), and adjusting the corresponding sliders. R: the radius of the path (in case of 0 it is straight), M: the mass of the AGV, Vm: is the maximal speed of the AGV. In editing mode obstacles are also can be generated (up to 5), by adjusting the "Obstacle" slider. The obstacles appeared this case can be moved by dragging them using the mouse (catch and move them by pressed left mouse button). The editing mode can be finished by pressing the "Ready" button. This case you can run the simulation again with the modified circumstances. The actual state of the system (the approximated necessities of the known partial strategies) is indicated on the four bottom sliders. P: is indicating the necessity of path tracking and collision avoidance, C: is the necessity of collision avoidance, R and L: are the necessities of collision avoidance with right/left tendency strategies. The actual state is adjusted by a fuzzy automata. Deselecting the "Auto" checkbox under the state sliders, the actual state can be adjusted manually by adjusting the corresponding state sliders (even during the simulation). Pressing the "Plot" button, when the simulation is stopped, the time function of the path, and the time functions of the observations, conclusions and system states of the previous simulation are appearing in separate windows. You can leave the program by pressing the "Close" button.

Some publications related to this program:

Sz. Kovács and L.T. Kóczy: "Application of the Approximate Fuzzy Reasoning Based on Interpolation in the Vague Environment of the Fuzzy Rulebase in the Fuzzy Logic Controlled Path Tracking Strategy of Differential Steered AGVs", Lecture Notes in Computer Science, 1226, pp.456-467. Springer, Germany, (1997).

Sz. Kovács and L.T. Kóczy: "Path Tracking and Collision Avoidance Strategy of an AGV Implemented on Interpolation-based Fuzzy Logic Controller", Proceedings of the INES'98 IEEE International Conference on Intelligent Engineering Systems, Vienna, Austria, pp.67-72, (1998).

Sz. Kovács, G. Terstyánszky, L. T. Kóczy and D. Vadász: "Similarity based System Reconfiguration in the control system of an experimental AGV", Proceedings of the SAFEPROCESS'2000, the 4th IFAC Symposium on Fault Detection and Diagnosis Supervision and Safety for Technical Processes, Budapest, 14-16 June, Vol. 1/2, pp.756-761, (2000).

Sz. Kovács: "Similarity based Control Strategy Reconfiguration by Fuzzy Reasoning and Fuzzy Automata", To appear in the proceedings of the IECON-2000, IEEE International Conference on Industrial Electronics, Control and Instrumentation, October 22-28, Nagoya, Japan, p.6, (2000).

Were you use any part of this program, please refer the source and some of these publications and send an e-mail to szkszilv@gold.uni-miskolc.hu about it.

The modules of the program (there are M.Files only):

AGVSIM.m: Calculate the next position for an AGV
AGVSYS.m: Calculate the next position for an AGV and the input parameters for the control
AGVSYSCA.m: Calculate the next position for an AGV and the input parameters for the control
CLRUNSP.m: Add new unsecure points and clean up the unsecure points matrix
EDIAGVSP.m: Switch to the edit the AGV simulation parameters mode
GAGVFLCA.m: Generate parmeters for Fuzzy Automata, for interpolation in vague environment
GAGVFLCC.m: Generate parmeters for Fuzzy Path Tracking and Collision Avoidance
GAGVFLCF.m: Generate parmeters for Collision Avoidance, with Symmetric, Right and Left Tendency
GAGVFLCM.m: Generate parmeters for the metacontrol.
GAGVFLCV.m: Generate parmeters for Fuzzy Path Tracking
GENAGV.m: Generate the AGV parameters.
GENAPATH.m: Generate the path parameters and the starting position
GENOBST.m: Generate the obstacle parameters
GEUNSECP.m: Generate unsecure points from two ultrasonic observations
GPATHP.m: Generate the points of the path
GSCFUNC.m: Generate the scaling function SCF
GSCLFUNC.m: Generate the linear scaling function SCF
GSCNFUNC.m: Generate the nonlinear scaling function SCF
GVAGENV.m: Generate vague environment VE
INIAGVSP.m: Initialize the AGV simulation parameters
INITAGVO.m: Init the AGV object
INITOBSO.m: Init the obstacle objects
MODPATHR.m: Modify the path radius
MOVEOBJ.m: Move the obstacle objects in edit mode
MOVEOBST.m: Init the obstacle objects
NORMVAR.m: Normalization and limitation of a variable
PATHDIST.m: Calculate the distance from the path
PLAGVLIN.m: Plot AGV lines (trajectories) on screen
PLAGVRES.m: Plot the results of the AGV simulation
PLAGVS.m: Plot AGVs on screen
PLOBST.m: Plot the obstacles
PLOTAGV.m: Plot an AGV on screen
PLOTPAGV.m: Plot the path and the AGV on the screen in its initial position
PNBDIST.m: Distance of a near by point from a line
REFROBST.m: Refresh the visibility of the obstacle objects
RPLOTAGV.m: Running AGV on screen
RULEDIST.m: Calculate the distances of the observation from the rule antecedents
RUNAGV.m: RUN the AGV simulation
SIDETCHA.m: Calculate the max turn angle without side touch (collision)
TOUCH.m: Calculate the collision to the obstacles
USENSE.m: Calculate the distances measured by the ultrasonic sensors
vagconcl.m: Calculate the conclusion from the observation and the rulebase
VAGDIST.m: Calculate the distance of two points in vague environment
VALVAG.m: Value of a vague point VP