MOTION CONTROL OF A CAR-TYPE ROBOT, SEEKING TO REACH A GIVEN POSITION

KAPITONOV A.A. (CONTROL SYSTEMS AND INFORMATICS DEPARTMENT, PhD)

ARTEMOV K.A., ANTONOV E.S. (P4235)

SUZDALEV O.D., AL-NAIM R.I., KARAVAEV A.A., ZAMOTAEV E.V. (P3235-36)

TASK DEFINITION

For a mobile robot with a structure ressembling that of a car:

- implement a motion control system that allows the robot to reach a given goal without colliding with any foreign object
- implement a computer vision system that determines the position of the surrounding objects and the robot itself
- implement a graphical user interface allowing to set the desired position of the robot and monitor its movement as it moves towards it

USED ROBOT

Fig. 1 Front view of the robot

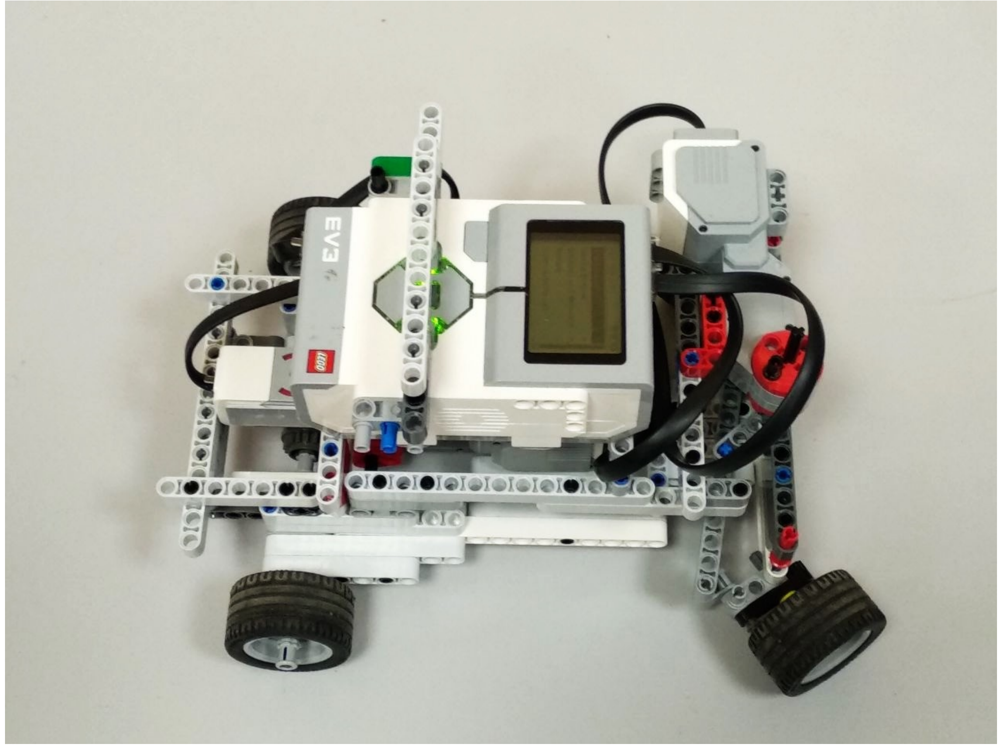


Fig. 2 View from above

MATHEMATICAL MODEL OF THE ROBOT

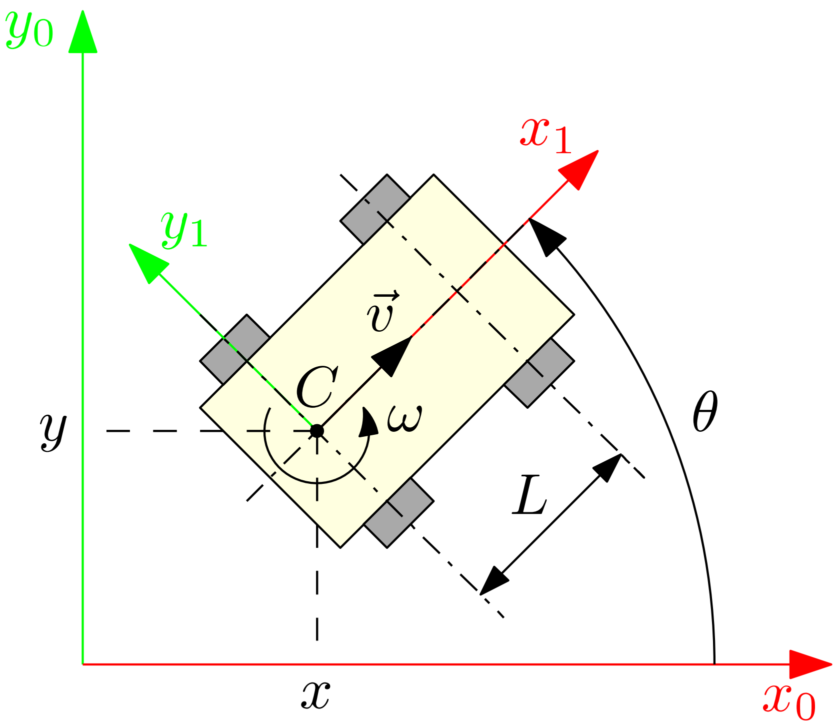


Fig. 3 Explanatory drawing

$$\begin{cases} \dot{x} = v \cdot \cos\theta \\ \dot{y} = v \cdot \sin\theta \\ \dot{\theta} = \omega = \dfrac{v}{L}\tan\overline{\varphi} \end{cases}$$

Few facts:

$Ox_0y_0 = Ox_1y_1$ for $t = 0$;
v and ω — inputs;
for $v \geq 0$ — forward motion;
for $v \leq 0$ — backward motion;

Physical meaning of each value:

x, y — coordinates of point C, middle of rear axle;
θ — rotation angle of the robot;
v — projection of velocity $\vec{v}$ (of the point C) on the Ox_1 axis;
ω — angular velocity of the robot;
$\overline{\varphi}$ — steering angle

B. PADEN, M. ČÁP, S. Z. YONG, D. YERSHOV AND E. FRAZZOLI, "A SURVEY OF MOTION PLANNING AND CONTROL TECHNIQUES FOR SELF-DRIVING URBAN VEHICLES," IN IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, VOL. 1, NO. 1, PP. 33-55, MARCH 2016.

MOTION CONTROL SYSTEM

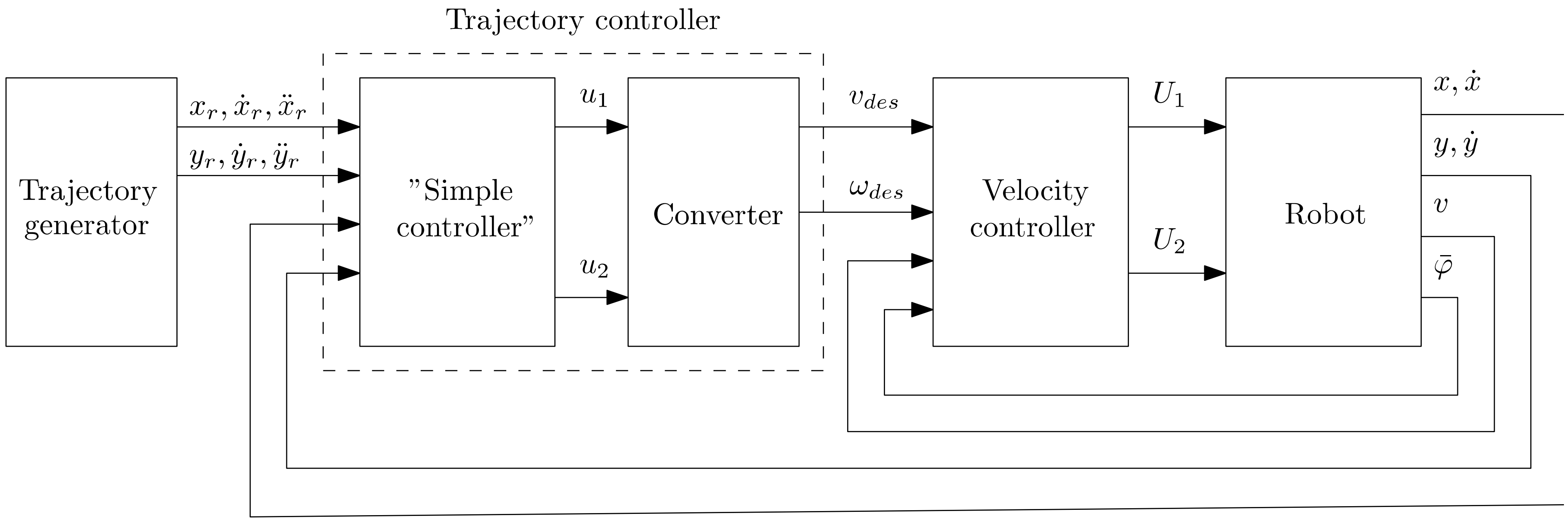


Fig. 4 General scheme of the robot motion control system

U_1 (U_2) — voltage applied to the traction (steering) engine, expressed as a percentage of the maximum voltage;

$\bar{\varphi}$ — the angle of the shaft of the steering engine;

x_r *and* y_r — coordinates that the robot must have at a given time to follow the desired path;

X_{des} — desired value of X

VELOCITY CONTROLLER

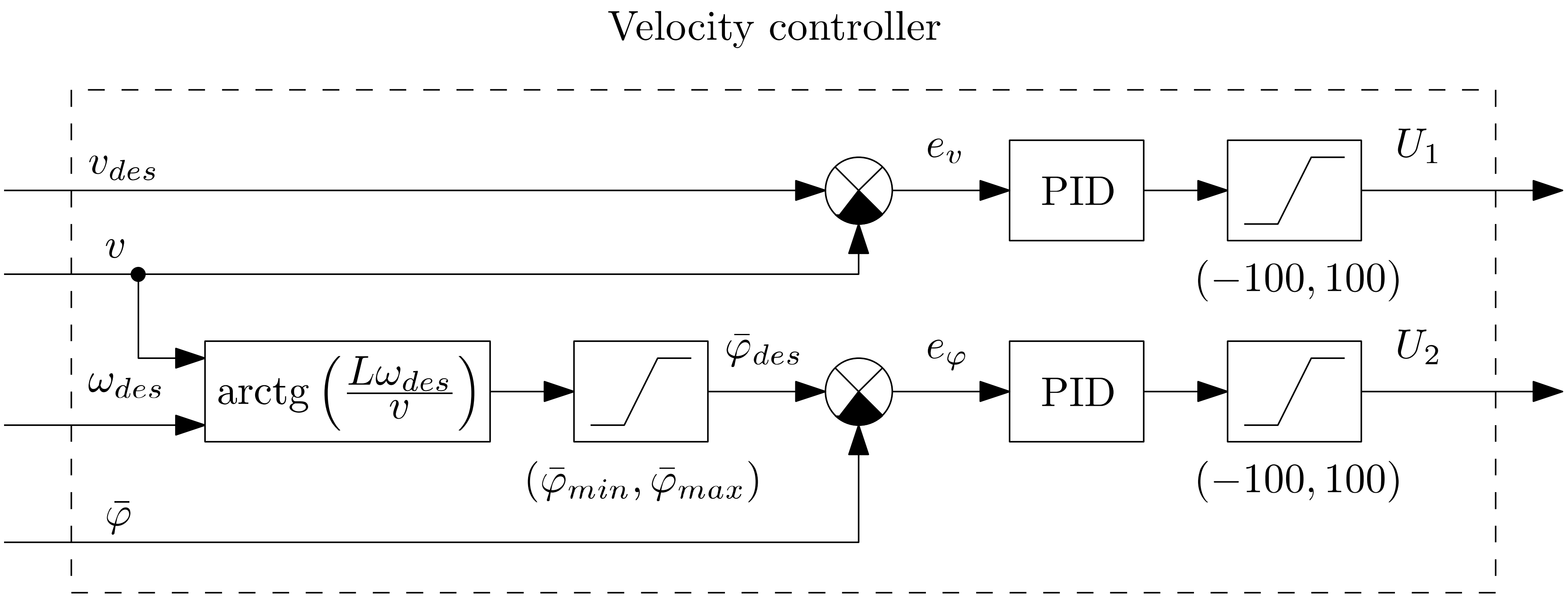


Fig. 5 Strucure scheme of the velocity controller for the robot

VELOCITY CONTROLLER

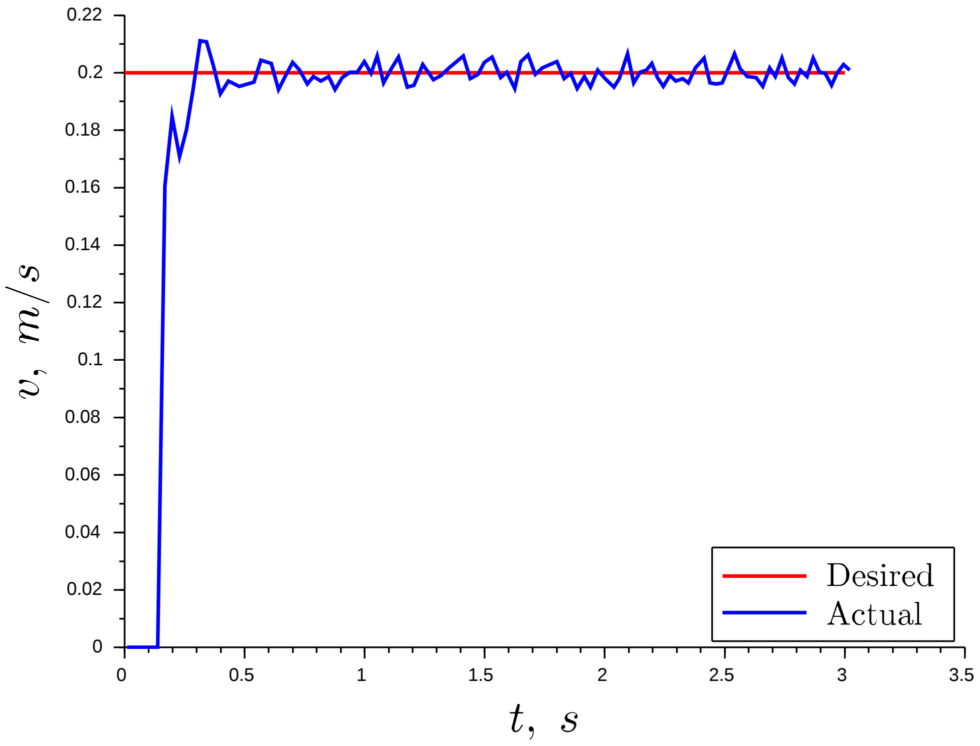


Fig. 6 Step response of the linear velocity controller

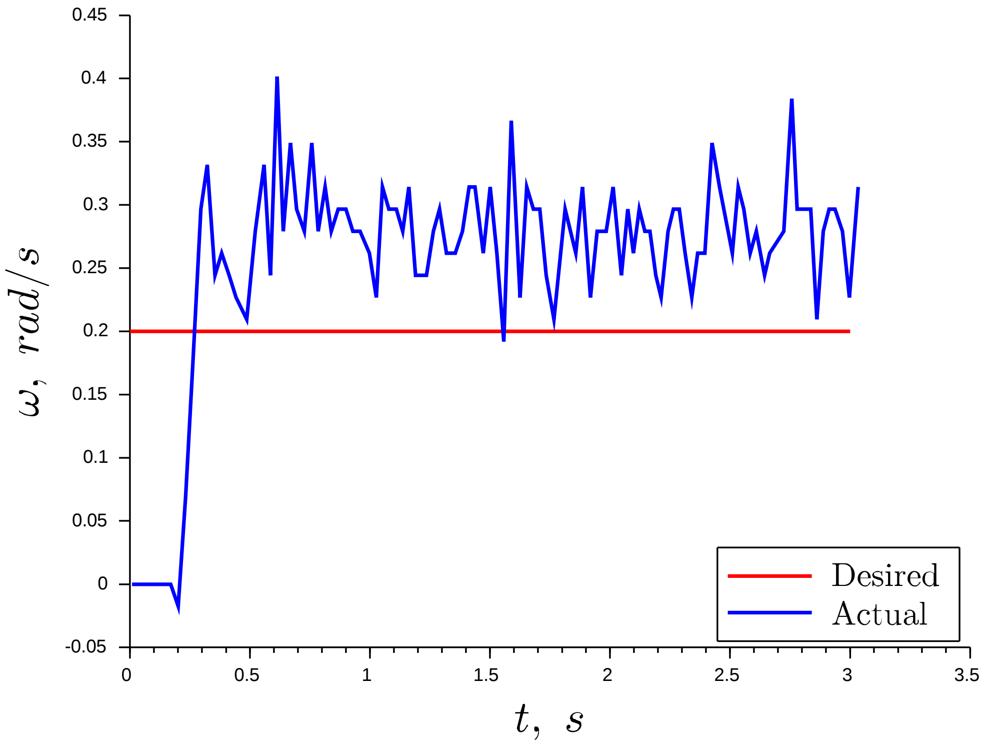


Fig. 7 Step response of the angular velocity controller

TRAJECTORY CONTROL

Linearization of the robot's mathematical model and control law

- $$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \omega \end{cases}$$
- $$\begin{cases} \dot{\xi} = u_1\cos\theta + u_2\sin\theta \\ v = \xi \\ \omega = \dfrac{-u_1\sin\theta + u_2\cos\theta}{\xi} \end{cases}$$
- $$\begin{cases} \ddot{x} = u_1 \\ \ddot{y} = u_2 \end{cases}$$
- $$\begin{cases} u_1 = \ddot{x}_r + k_{p1}(x_r - x) + k_{d1}(\dot{x}_r - \dot{x}), \\ u_2 = \ddot{y}_r + k_{p2}(y_r - y) + k_{d2}(\dot{y}_r - \dot{y}) \end{cases}$$

Formulas in practical implementation

- $$\begin{cases} \dot{\xi} = u_1\cos\theta + u_2\sin\theta\,, \\ v_{des} = \bar{\xi}, \\ \omega_{des} = \bar{\omega}_{des}, \end{cases}$$
- $$\bar{\omega}_{des} = \begin{cases} \dfrac{-u_1\sin\theta + u_{des}\cos\theta}{\bar{\xi}}, \bar{\xi} \neq 0 \\ 0,\ \bar{\xi} = 0 \end{cases}$$
- $$\bar{\xi} = \begin{cases} \xi,\ \bar{\xi} \in (-v_{\max}, v_{\max}) \\ v_{\max}, \xi \geq v_{\max} \\ -v_{\max},\ \bar{\xi} \leq v_{\max} \end{cases}$$

ALESSANDRO DE LUCA, GIUSEPPE ORIOLO, MARILENA VENDITTELLI, "STABILIZATION OF THE UNICYCLE VIA DYNAMIC FEEDBACK LINEARIZATION", IFAC PROCEEDINGS VOLUMES, VOLUME 33, ISSUE 27, 2000, PAGES 687-692

TRAJECTORY CONTROLLER

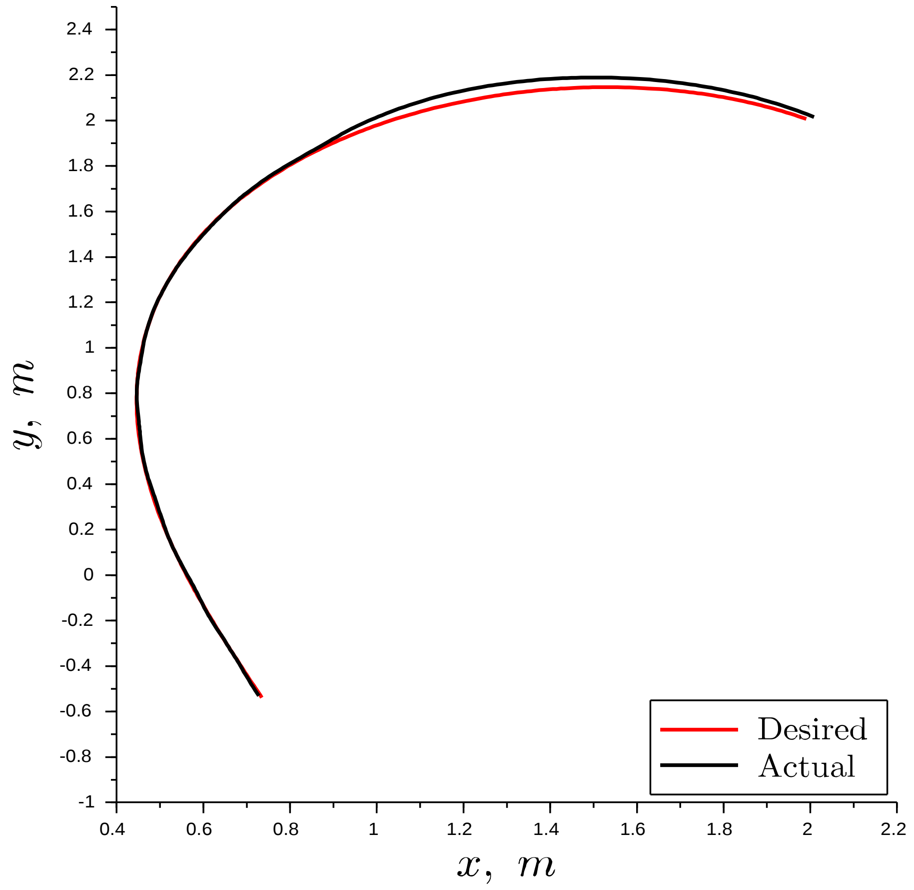


Fig. 8 Test result of the robot for clothoid type trajectory

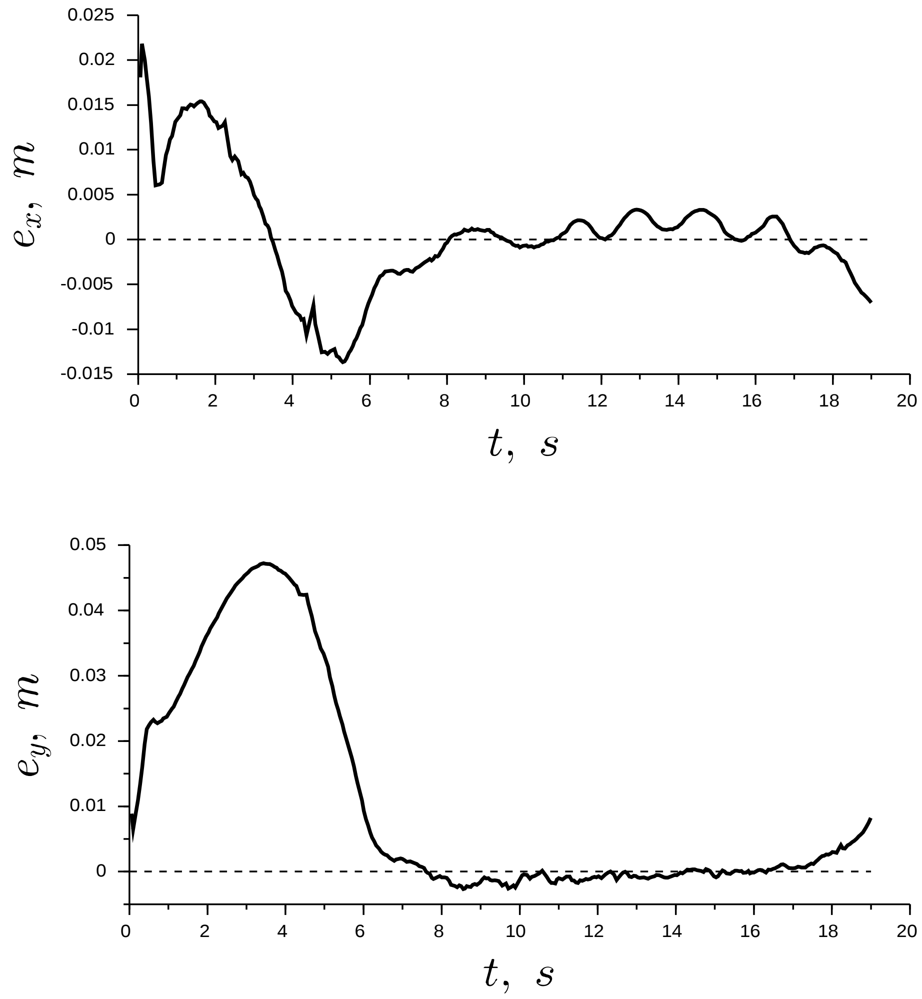


Fig. 9 Control errors

TRAJECTORY PLANNING

Fig. 10 Illustration of path-finding towards goal

DOMOKOS KISS AND GÁBOR TEVESZ, “AUTONOMOUS PATH PLANNING FOR ROAD VEHICLES IN NARROW ENVIRONMENTS: AN EFFICIENT CONTINUOUS CURVATURE APPROACH,” JOURNAL OF ADVANCED TRANSPORTATION, VOL. 2017, ARTICLE ID 2521638, 27 PAGES, 2017.

TRAJECTORY PLANNING

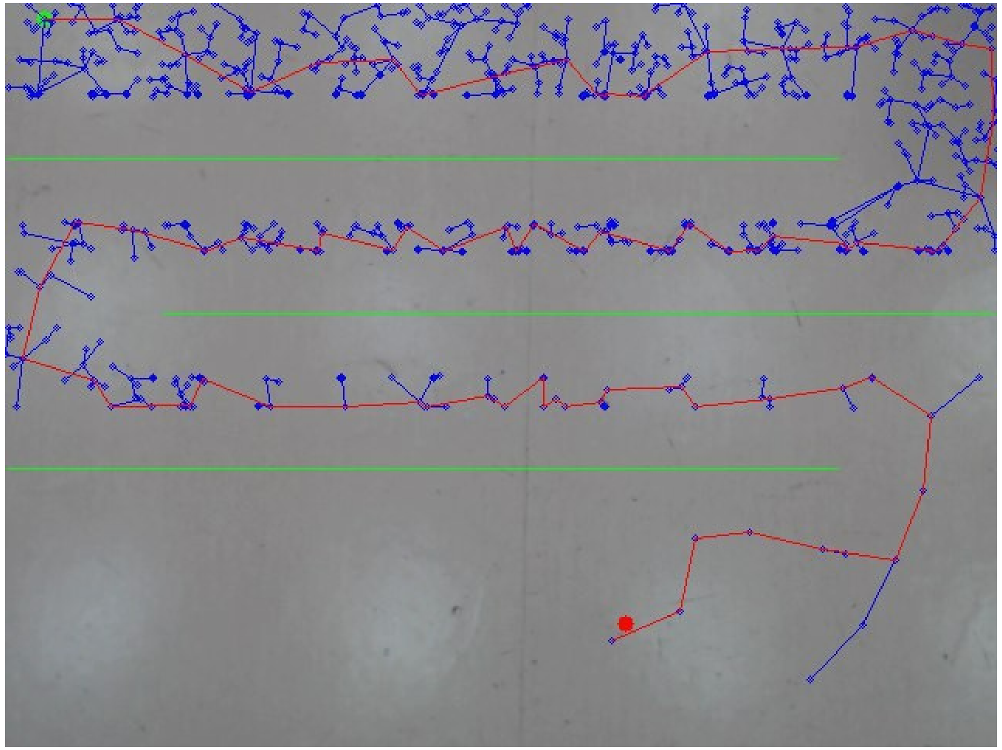


Fig. 11 The result of the RRT algorithm

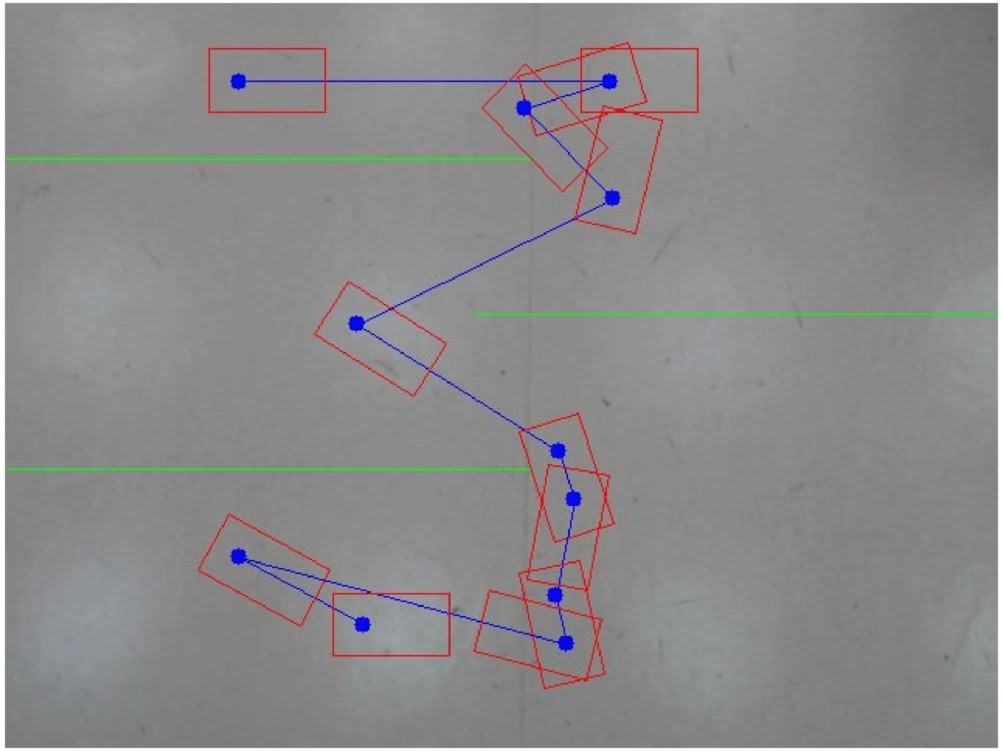
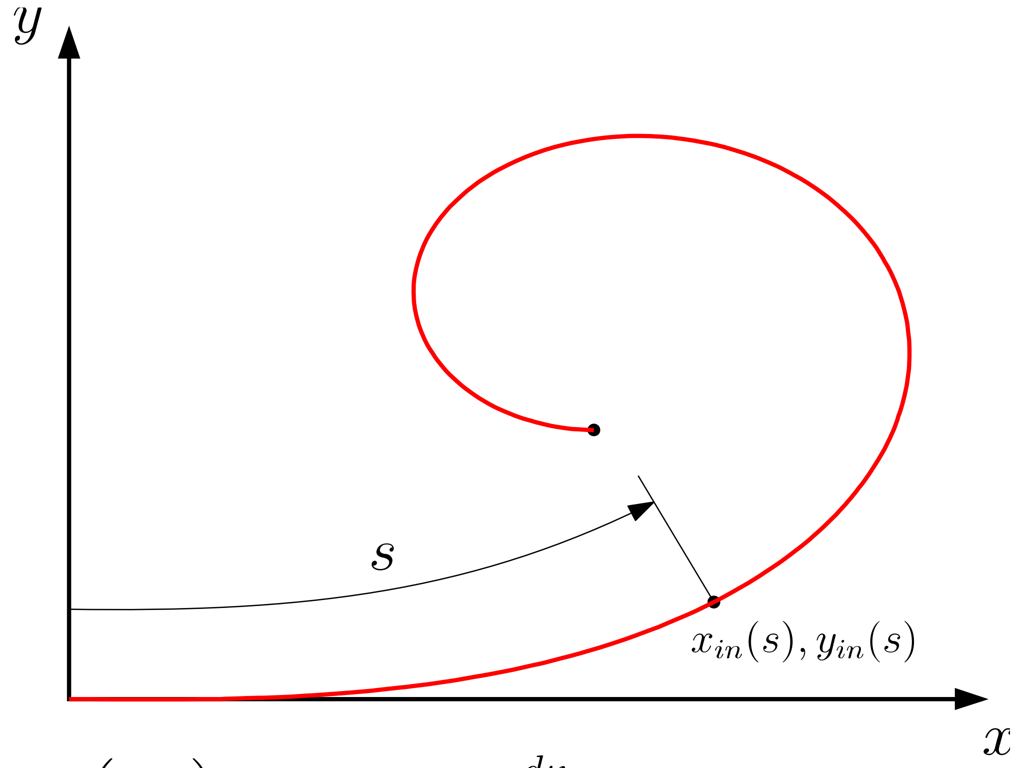


Рис. 12 The result of the RTR algorithm

TRAJECTORY PLANNING

$$\begin{cases} x_{in}(s) = \gamma\sqrt{\dfrac{\pi}{|\alpha|}} \cdot C_F\left(\sqrt{\dfrac{|\alpha|}{\pi}}s\right), \\ y_{in}(s) = \gamma \ \mathrm{sign}\,\alpha\sqrt{\dfrac{\pi}{|\alpha|}} \cdot S_F\left(\sqrt{\dfrac{|\alpha|}{\pi}}s\right) \end{cases}$$

$$\begin{cases} x_{out}(s) = -\gamma\sqrt{\dfrac{\pi}{|\alpha|}} \cdot C_F\left(\sqrt{\dfrac{|\alpha|}{\pi}}(2s_{end} - s)\right), \\ y_{out}(s) = \gamma \ \mathrm{sign}(\alpha)\sqrt{\dfrac{\pi}{|\alpha|}} \cdot S_F\left(\sqrt{\dfrac{|\alpha|}{\pi}}(2s_{end} - s)\right) \end{cases}$$



$$\begin{cases} C_F(x) = \int\limits_0^x \cos\left(\dfrac{\pi}{2}\mu^2\right)\, d\mu \approx \dfrac{1}{2} + f(x)\sin\left(\dfrac{\pi}{2}x^2\right) - g(x)\cos\left(\dfrac{\pi}{2}x^2\right), \\ S_F(x) = \int\limits_0^x \sin\left(\dfrac{\pi}{2}\mu^2\right)\, d\mu \approx \dfrac{1}{2} - f(x)\cos\left(\dfrac{\pi}{2}x^2\right) - g(x)\sin\left(\dfrac{\pi}{2}x^2\right), \\ f(x) = \dfrac{1 + 0.926x}{2 + 1.792x + 3.104x^2}, \qquad g(x) = \dfrac{1}{2 + 4.142x + 3.492x^2 + 6.670x^3} \end{cases}$$

$\alpha = \dfrac{d\kappa}{ds} = const$

$\kappa = 1/\rho$ — curvature

ρ — radius of curvature

$\gamma \in \{-1; 1\}$

D. K. WILDE, "COMPUTING CLOTHOID SEGMENTS FOR TRAJECTORY GENERATION," 2009 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, ST. LOUIS, MO, 2009, PP. 2440-2445.

Fig. 13 Clothoid appearance

TRAJECTORY PLANNING

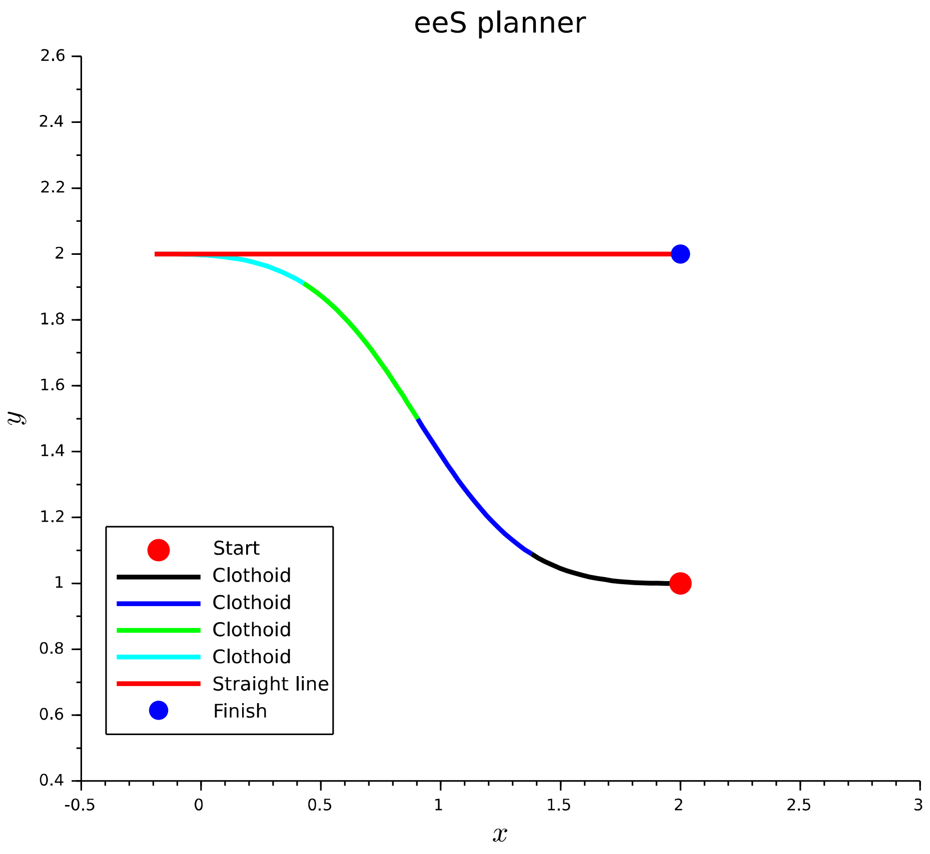


Рис. 14 Trajectory from eeS planner

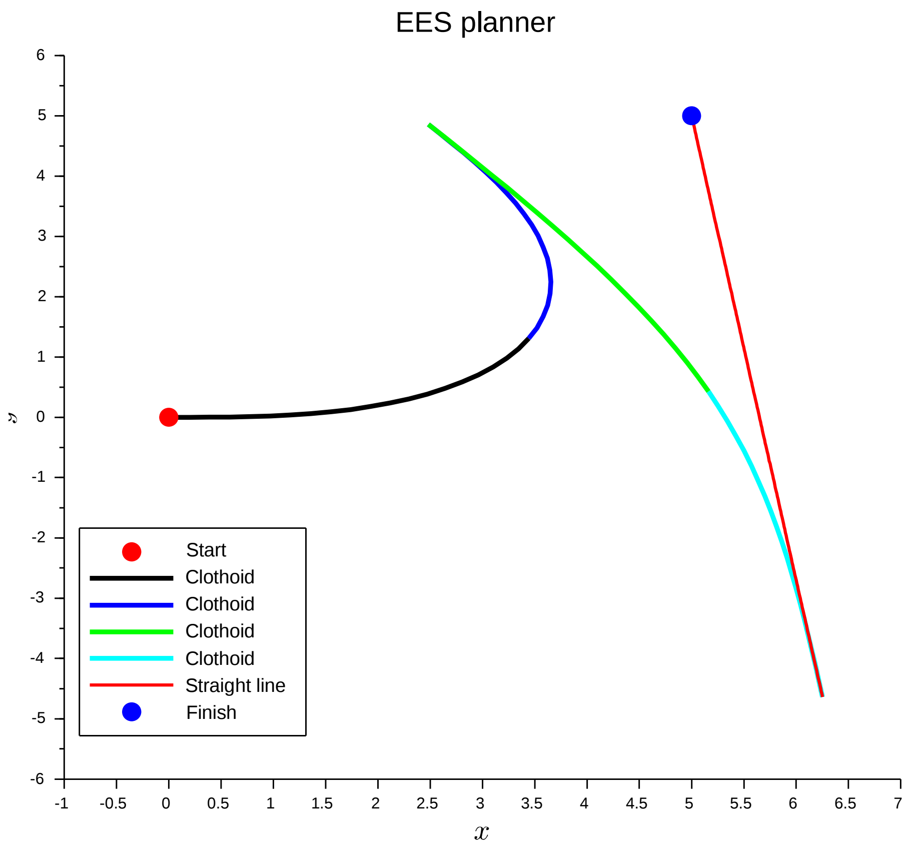


Рис. 15 Trajectory from EES planner

TRAJECTORY PLANNING

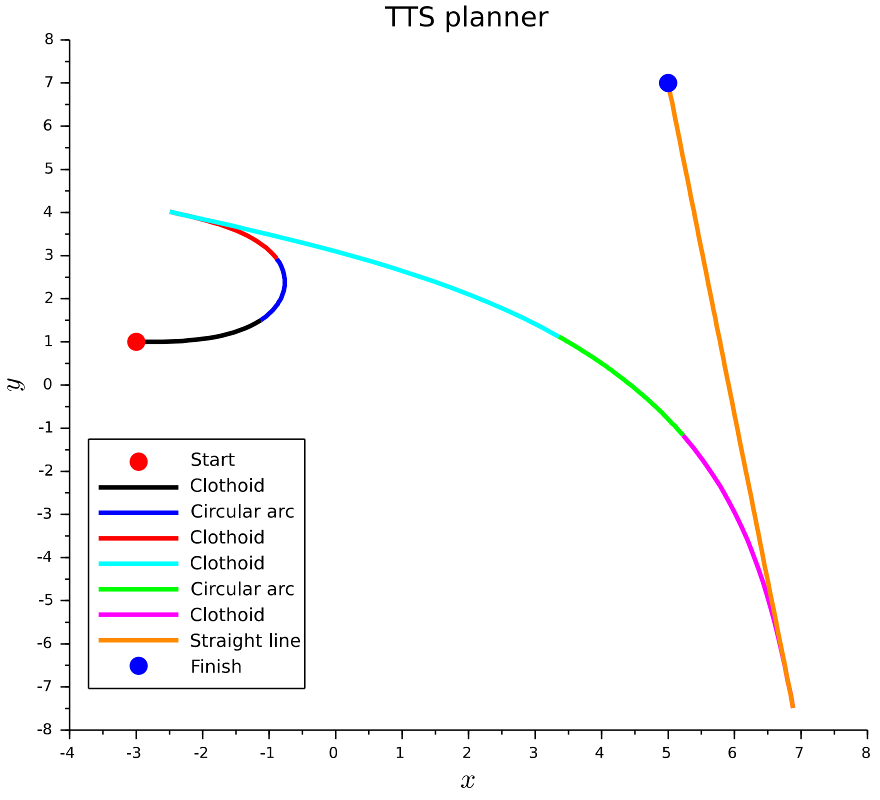


Рис. 16 Trajectory from TTS planner

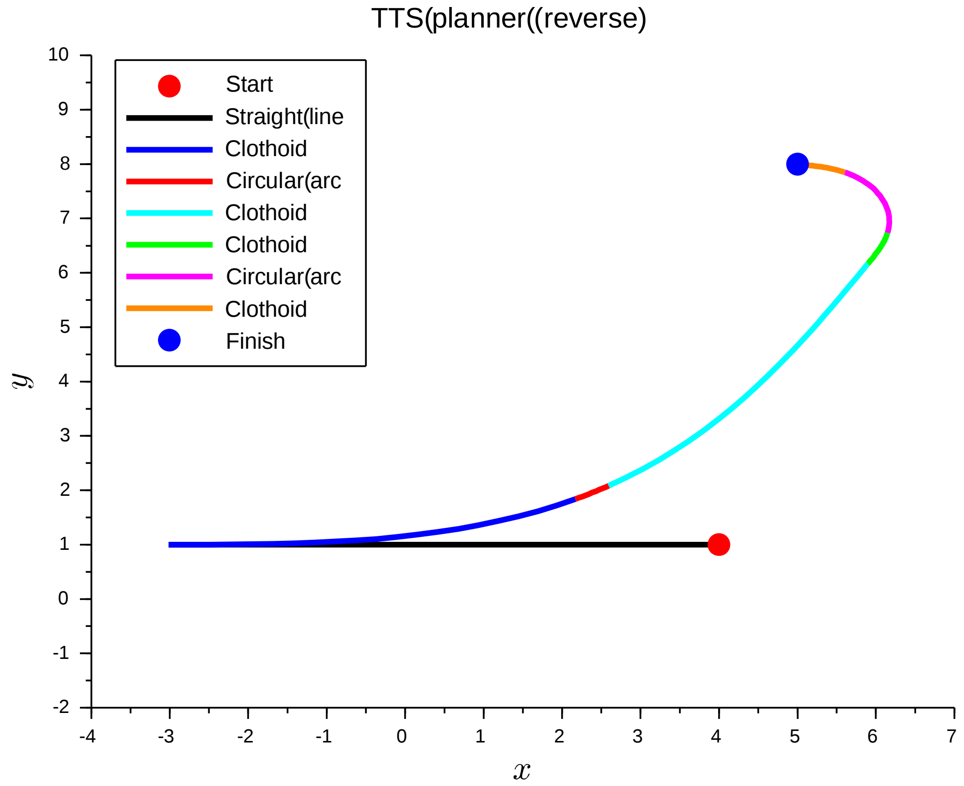


Рис. 17 Trajectory from TTS planner in reverse search direction

COMPUTER VISION SYSTEM

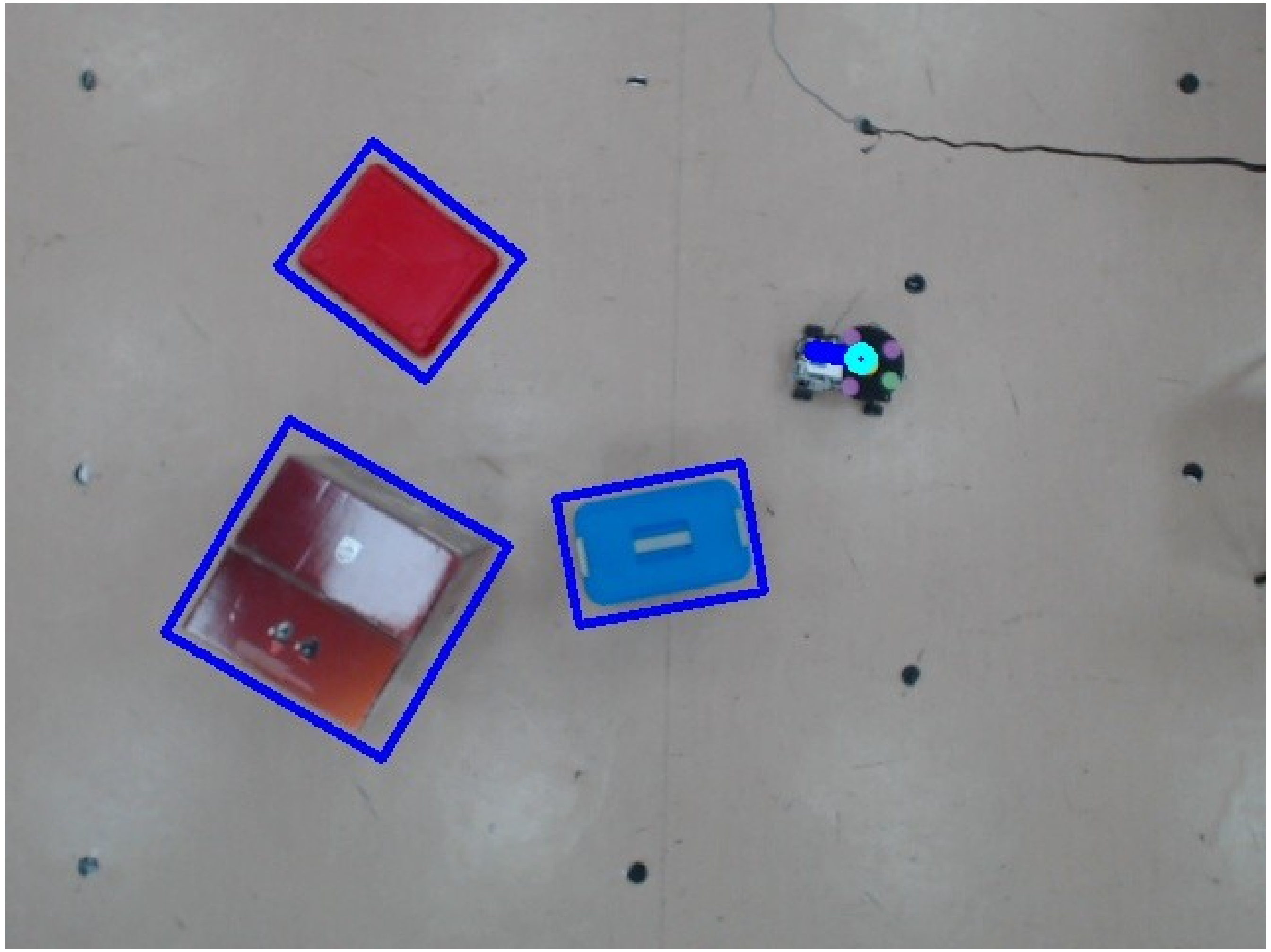


Fig. 18 Result of the localization of the robot and obstacles by the computer vision algorithm

GRAPHICAL USER INTERFACE

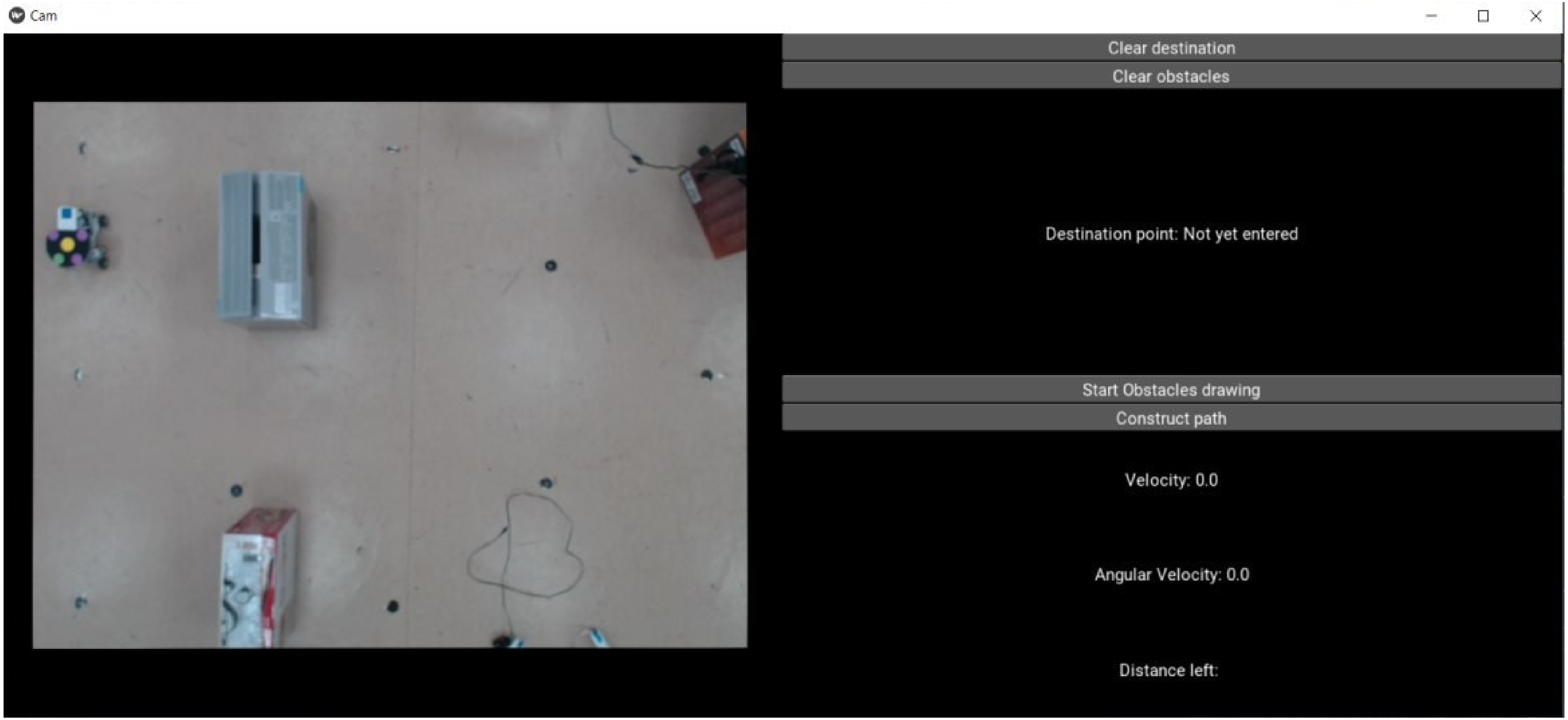


Fig. 19 Appearance of the developed application

GRAPHICAL USER INTERFACE

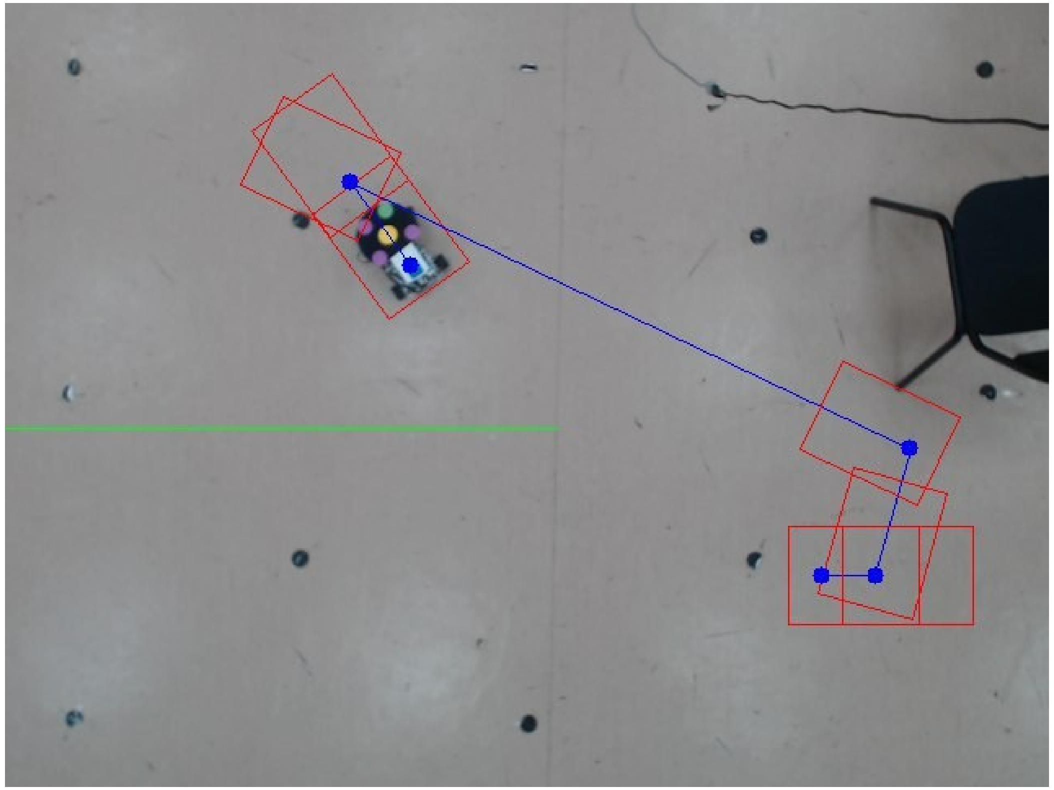


Fig. 20 Global path visualization

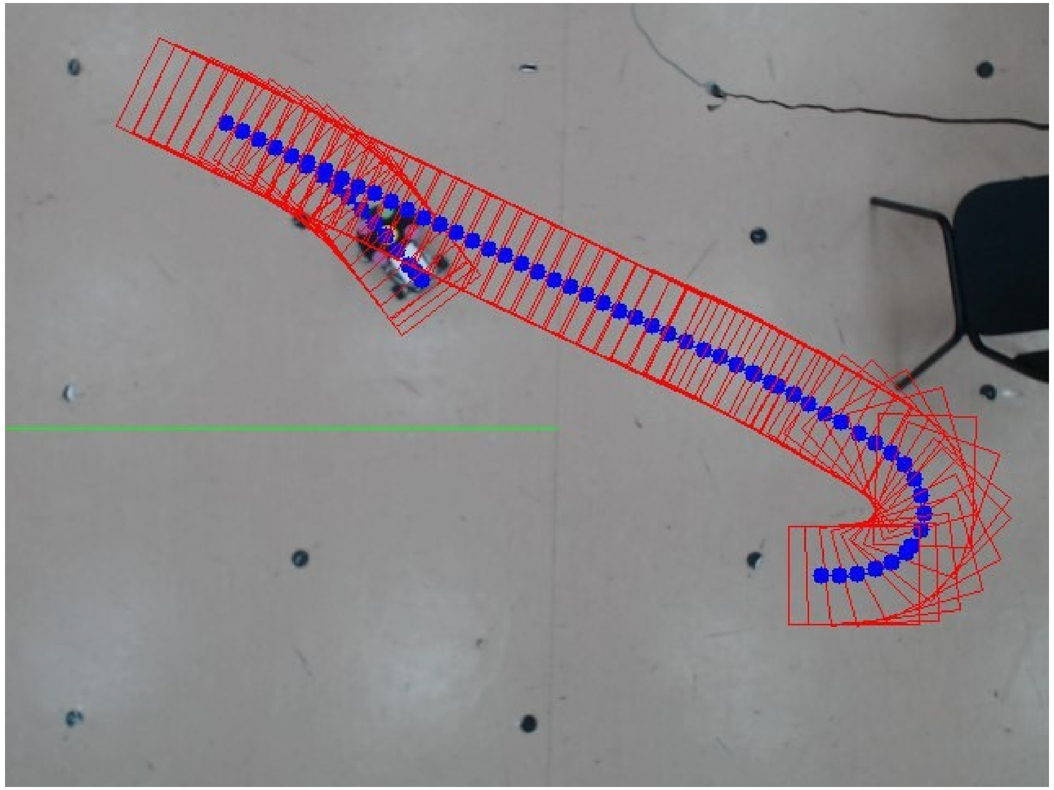


Fig. 21 Visualization of path approximation

CONCLUSION

- What we did

- Developed a robot motion control system
- Probably the first in the world to have implemented RTR and TTS planners on a real robot
- Developed a computer vision system that performs navigation tasks
- Developed a control application with GUI

- What can be done in the future

- Improve the quality of angular velocity control
- Improve acceleration and braking operations of the robot
- Increase stability of the system
- Convert to ROS

THANK YOU