

# Stochastic Multi-objective Optimization for Aggressive Maneuver Trajectory Planning on Loose Surface <sup>★</sup>

Alexander V. Terekhov\* Jean-Baptiste Mouret\*\*  
Christophe Grand\*\*\*

\* *Institut Systèmes Intelligents et de Robotique, UPMC-CNRS,*

*4 Place Jussieu, 75005 Paris, France*

*e-mail: avterekhov@gmail.com*

*\*\* email: mouret@isir.upmc.fr*

*\*\*\* e-mail: christophe.grand@isir.upmc.fr*

---

**Abstract:** In this work we regard the problem of trajectory planning for aggressive maneuver of a wheeled mobile robot on loose surface. Our approach is inspired by previously reported analysis of professional rally racers actions during sharp turn. Using numerical simulations we obtain a set of solutions representing all possible compromises between speed and accuracy of maneuver. We choose a particular solution and extend it by producing a continuous mapping from desired trajectory turn angles to parameters of the control inputs.

*Keywords:* autonomous mobile robots, trajectory planning, stochastic optimization

---

## 1. INTRODUCTION

Autonomous field robotics is an attractive and challenging area of research. The problems range from obstacles avoidance, path planning and trajectory tracking under space limitations, to enhancement of the obstacle climbing capabilities on highly uneven terrain. The most famous application, of such researches is the planetary exploration. In most cases, the considered velocities are low and involve only small dynamical perturbations on the system. Thus, most of the developed approaches are based on kinematic modeling. Since recently research projects started to analyze problems induced by the locomotion of ground vehicle at higher velocity (about 10 m/s). One of the famous milestone among past researches was the Great DARPA Challenge that took place in 2005 (see Buehler et al. (2007)). This kind of challenge requires to find a way to deal with the problems, like locomotion on natural terrain, the control of highly dynamic behavior, the presence of large wheel-ground slippage, problems of perception at high rate frequency, the planning of feasible reference trajectory, etc.

In this paper we focus on the problem of path planning on slipping terrain that allows for aggressive maneuvers at high velocity. The general approach of our work is to control a wheeled mobile robot at high velocity, performing steering maneuvers by taking into account the wheel-ground slippage. We expect that admitting significant slippage angle during vehicle displacement would allow to noticeably increase the velocity of the maneuver.

In a past work, we have proposed methods to stabilize the control of a mobile robot performing a path tracking task on a slipping terrain (see Lucet et al. (2008b), however

see also Peters and Iagnemma (2008)). In this method the robot was driven by standard kinematic controller, but the final control inputs were corrected by an algorithm that stabilizes the yaw dynamics. The method has been extended to skid-steering robots using a sliding mode controller (see Lucet et al. (2008a)). This method allows to deal with the slippage when the stabilized trajectory is produced by a classical controller. An evident limitation of that work is that it does not allow to take advantage of slippage effect in order to produce aggressive maneuvers like over-steer. In this work, we propose a methodology that deals with the problem of feasible dynamic trajectory planning, i.e., that is consistent with the dynamics of the vehicle.

Recently few attempts were made to approach the problem of highly maneuver steering by getting inspiration from the performance of professional rally drivers (see Velenis et al. (2007a,b)). The driver's actions were recorded during a sharp steering maneuver with 90 degrees turning angle. The authors proposed a reasonable approximation of the driver's actions: the steering, the break and the throttle were shown to be nearly piecewise linear functions of time. Using a computer model the authors have shown that such approximation produces vehicle motion similar to the one exhibited by rally racers.

In order to use the solutions proposed by Velenis et al. (2007a) as a reference trajectory for the vehicle it is necessary to have a set of possible maneuvers including solutions for different trajectory turn angle, road conditions, etc. In fact, one needs a mapping from the set of task conditions to the set of corresponding maneuvers. It is highly desirable that such mapping is sufficiently smooth. Indeed, it may happen, for example, that in course of a maneuver the initial estimation of the road turn angle

---

<sup>★</sup> The work was funded by DGA, grant REI 2008.34.0018

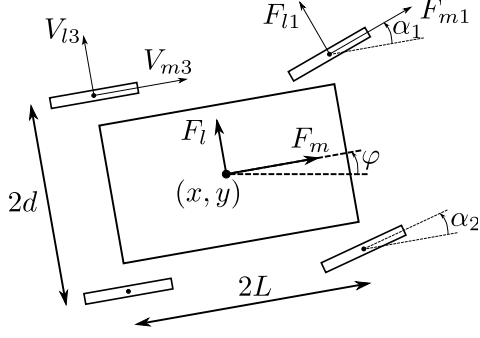


Fig. 1. Schematic representation of the robot.

is discovered to be incorrect and thus another maneuver should have been used. In case of continuous mapping the discrepancy between the performed and the necessary maneuver is not so dramatic and thus, a correction can be made. In addition, the continuous mapping could be used for interpolation of solutions absent in the initial set. The ability of skilled human rally drivers to perform maneuver in different conditions make us think that such a continuous mapping may be possible.

In the current study we aim towards adapting the solution proposed in Velenis et al. (2007a) for a mobile robot with two independent steering wheels and four driving wheels. We want to extend the obtained solution for a range of the angles of the road turn and thus to obtain a continuous mapping from the turn angles to the parameters of the control inputs. For this purpose, we first obtain a mathematical model of the robot, then find a solution for a given road turn angle, e.g 90 degrees, and finally extend the solution continuously for other turn angles. To obtain the solutions we use algorithms of stochastic optimization.

## 2. MATHEMATICAL MODEL

For our purpose we need a computationally inexpensive model of a mobile robot, which at the same time captures the main features of its behavior. Currently we are only interested in the robot's movement when all four wheels stay in contact with the ground. For the sake of simplicity we regard only planar motion of the robot (see Fig. 1).

The position of the robot can be defined by the location of its center of mass  $x, y$  and the heading angle  $\varphi$  between the  $x$  axis and the longitude axis of the robot. The motion of the robot satisfies the equations:

$$\begin{aligned}
 \dot{x} &= V_m \cos \varphi - V_l \sin \varphi \\
 \dot{y} &= V_m \sin \varphi + V_l \cos \varphi \\
 \dot{\varphi} &= \omega \\
 J\dot{\omega} &= T \\
 M\dot{V}_m &= F_m + M\omega V_l \\
 M\dot{V}_l &= F_l - M\omega V_m
 \end{aligned} \tag{1}$$

where  $\omega$  is the angular velocity of the robot's trunk,  $V_m$  and  $V_l$  are the projections of the linear velocity of the robot's center of mass on the medial (longitude) and lateral axes respectively,  $F_m, F_l$  and  $T$  are the total medial and lateral forces and the total torsion torque, defined as the following:

$$\begin{aligned}
 F_m &= F_{m1} \cos \alpha_1 - F_{l1} \sin \alpha_1 + F_{m2} \cos \alpha_2 - \\
 &\quad F_{l2} \sin \alpha_2 + F_{m3} + F_{m4} \\
 F_l &= F_{l1} \cos \alpha_1 + F_{m1} \sin \alpha_1 + F_{l2} \cos \alpha_2 + \\
 &\quad F_{m2} \sin \alpha_2 + F_{l3} + F_{l4} \\
 T &= L(-F_{m1} \cos \alpha_1 + F_{l1} \sin \alpha_1 + F_{m2} \cos \alpha_2 - \\
 &\quad F_{l2} \sin \alpha_2 - F_{m3} + F_{m4}) + \\
 &\quad d(F_{l1} \cos \alpha_1 + F_{m1} \sin \alpha_1 + F_{l2} \cos \alpha_2 + \\
 &\quad F_{m2} \sin \alpha_2 - F_{l3} - F_{l4})
 \end{aligned}$$

$F_{mi}, F_{li}$  are lateral and medial projections of the tangential forces of wheel-road interaction for each wheel reference frame (see Fig. 1),  $\alpha_1, \alpha_2$  are the steering angles of the left and right wheels, respectively.

For the forces  $F_{mi}$  and  $F_{li}$  we use the brush model, which is relatively simple computationally, but at the same time captures main features of the wheel-road interaction. The details of the brush model can be found in Pacejka (2005). Roughly, the medial and lateral tangential forces are defined as nonlinear function of the lateral and medial projections of the slippage velocity,  $V_{sli}$  and  $V_{smi}$ , that is the velocity of the contact point of the wheel (in case of no sliding this velocity is zero):

$$\begin{aligned}
 F_{mi} &= \mu F_{ni} f(V_{smi}/V_i) \\
 F_{li} &= \mu F_{ni} f(V_{sli}/V_i)
 \end{aligned}$$

where  $V_i$  is absolute value of the velocity of the center of  $i$ -th wheel,  $F_{ni}$  is the normal force at the  $i$ -th wheel contact point and  $\mu$  is the coefficient of Coulomb friction. The function  $f$  depends on the tangential stiffness of the tire  $c_p$ .

To determine the normal forces we used the method described in Velenis et al. (2007b). The resultant normal forces for  $i$ -th wheel is given by the equations:

$$F_{ni} = \frac{M}{4dL} (dLg + hL_i \dot{V}_l + hd_i \dot{V}_m) \tag{2}$$

where  $h$  is the height of the center of mass of the robot.

The equation (2) describes weight redistribution caused by the acceleration of the center of mass of the robot in case when the pitch and roll angles of the robot are close to zero. However, to compute the medial and lateral acceleration one must provide the total medial and lateral tangential forces, which, according to the brush model, depend on the normal forces themselves. In order to solve this problem we made an assumption that the weight redistribution (2) does not happen instantly but with a characteristic time  $\tau$ , which roughly correspond to the characteristic time of the suspension system of the robot. We appended the dynamic equations of the robot (1) with the following:

$$\begin{aligned}
 \tau \dot{a}_l &= F_l/M - a_l \\
 \tau \dot{a}_m &= F_m/M - a_m
 \end{aligned}$$

and used the values  $a_l, a_m$  instead of  $\dot{V}_l, \dot{V}_m$  in the equation (2).

For sake of simplicity, we ignored the dynamics of the wheels and assumed that the velocity controller in each wheel tracks the desired wheel rotation velocity perfectly. We make the same assumption regarding the functioning of the steering system. Thus, 2 steering angles and 4 wheel rotation velocities represent the control inputs to the model. In order to reduce the dimension of the control inputs we regarded a steering angle  $\alpha$  of the front wheels

Table 1. Parameters of the model.

M	40 kg
J	3 kg·m <sup>2</sup>
L	0.50 m
d	0.25 m
h	0.10 m
$\mu$	0.6
$c_p$	10 <sup>5</sup> N/m <sup>2</sup>
$\tau$	0.05 s

and the linear velocities of the front and rear wheels,  $V_F$ ,  $V_R$ , correspondingly. The steering angles and velocities of the wheels were computed using Ackermann rule:

$$\begin{aligned} R\omega_1 &= V_F \left[ \left( 1 - \frac{d}{2L} \sin \alpha \right)^2 + \sin^2 \alpha \right]^{1/2} \\ R\omega_3 &= V_R \left( 1 - \frac{d}{2L} \sin \alpha \right) \\ \alpha_1 &= \arctan \frac{\sin \alpha}{1 - \frac{d}{2L} \sin \alpha} \end{aligned} \quad (3)$$

where  $\omega_1$ ,  $\omega_3$  are the left front and left rear wheel angular velocities;  $\alpha_1$  is the steering angle of the left front wheel;  $R$  is the wheel radius. The formulas for right wheels and steering angle are the same as (3), with the exception that every “plus” is substituted with “minus”.

The parameters of the model used in the simulations are presented in Tab. 1.

### 3. 90 DEGREES TURN

We first regard the case of 90 degrees turn. We assume that the desired path is composed of three parts: two linear parts orthogonal to each other and one circular part connecting them (see Fig. 4). The robot is located 30 m from the point of turn, directed towards that point and has initial velocity of 10 m/s. The simulation is run for 10 seconds, during which the robot is controlled in a feedforward manner. The goal is to perform a maneuver, minimizing the deviation from the desired path and maximizing the average movement velocity. These two objectives are clearly conflicting. To deal with the conflicting objectives, one usually fixes extremal value of one of them, for example, the maximum deviation, at a reasonable value and then solves the optimal problem with single objective and the additional inequality constraint. A significant drawback of this approach is that the choice of the extrema value is to great extent arbitrary, that provokes questions like how good the choice is and how the result of optimization depends on it. Another way to approach this problem is to use the framework of the multiobjective optimization.

Recent research in the stochastic optimization proposed numerous algorithms to simultaneously optimize several objectives (see Deb (2001)); most of them rely on the concept of Pareto dominance, defined as follows:

A solution  $p^*$  is said to dominate another solution  $p$ , if both conditions 1 and 2 are true:

- (1) the solution  $p^*$  is not worse than  $p$  with respect to all objectives;

- (2) the solution  $p^*$  is strictly better than  $p$  with respect to at least one objective.

The non-dominated set of the entire feasible search space is the globally Pareto-optimal set (Pareto front). It represents the set of optimal *trade-offs*, that is solutions that cannot be improved with respect to one objective without decreasing their score with respect to another one.

Pareto-based multi-objective optimization algorithms aim at finding the best approximation of the Pareto front, both in terms of distance to the Pareto front and of uniformity of its sampling. The whole set of optimal trade-offs is found using only one execution of such algorithm and the choice of the final solution is left to the researcher/engineer. Compared to classic approaches to multi-objective optimization, which most of the time aggregate objectives (e.g. with a weighted sum) then rely on a single-objective optimization algorithm, multi-objective optimization algorithm don't require to tune the relative importance of each objective. Moreover, depending of the shape of the Pareto front, some optimal trade-offs cannot be found using the most classic aggregation methods (see Deb (2001)).

Typical algorithms of multiobjective optimization first generate a set of  $N$  random points, called a population. Then they enter a loop of four steps until a convergence criteria is met (in this work, a fixed number of iterations is performed):

- (1) Sort population with respect to dominance such that non-dominated candidate solutions are ranked 1, those which are only dominated by non-dominated ones ranked 2, etc. Candidate solutions that are attributed the same rank are then sorted with regard to a diversity measure (in objective space) to favor solutions in the less crowded parts.
- (2) Keep only the best  $N$  solutions (during the first iteration, this step is useless).
- (3) Use the sorted population to generate new candidate solutions by perturbing the kept ones (e.g. by adding a Gaussian noise).
- (4) Merge the newly generated candidate solutions and the previous population; this gives the new population.

In this work, we employed NSGA-2 (see Deb et al. (2000)), one of the most efficient stochastic multi-objective optimization algorithm (see Zitzler et al. (2000)); it follows the previously introduced steps. The computations were performed using the Spheres\_v2 framework described in Mouret and Doncieux (2010).

Solving the mentioned optimization problem in the space of all possible control functions is barely possible. So, we used piecewise linear functions of time, similar to those proposed in Velenis et al. (2007a). We parametrized the time change of the steering angle  $\alpha$ , the velocity of the front wheels  $V_F$  and the velocity of the rear wheels  $V_R$  with the duration of each linear part and the amplitude of its change. Schematic representation for the steering angle is given in Fig. 2. The velocities were allowed to change within the range from 1 to 10 m/s, the steering angle varied from  $-60$  to  $60$  degrees. The time intervals were forced to be not longer than 5 s and not shorter than 0.4 s. In addition, they were checked to not exceed 10 s interval

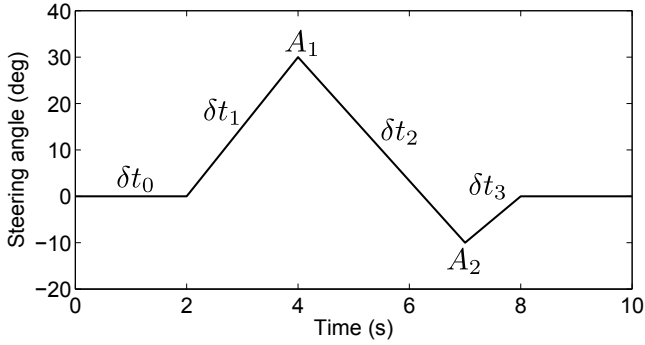


Fig. 2. Schematic illustration of parametrization used for the steering angle control input.

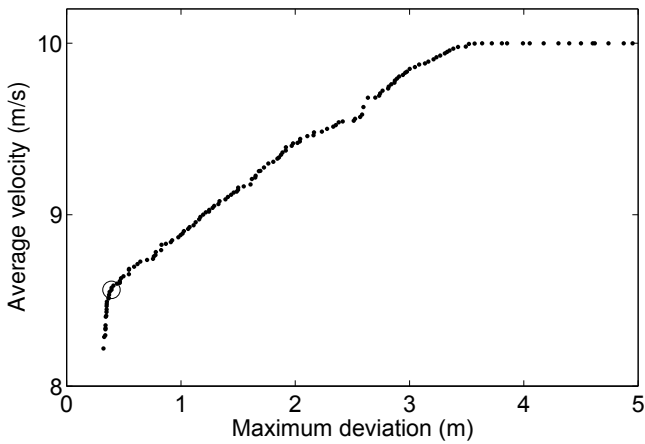


Fig. 3. Example of a Pareto front for 90 degrees turn. The circle denotes the selected solution on the front.

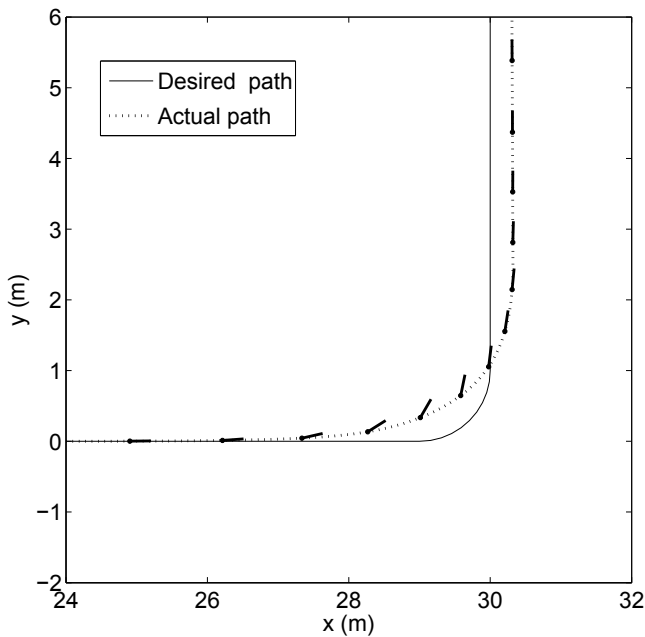


Fig. 4. The desired and actual trajectory for the selected solution. Black circles denote robot position every 250 ms; the straight lines show the direction of the robot.

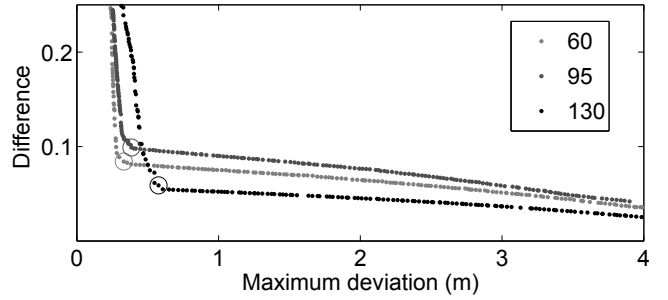


Fig. 5. Examples of a Pareto front for different angles. The circles denote the selected solutions.

on the whole. For the velocities  $V_F$  and  $V_B$  the initial and final values were set to 10 m/s, for the steering angle they were set to 0 degrees. We had 6 parameters for each control input (see Fig. 2) and 18 parameters on the whole.

The estimated Pareto front is provided in Fig. 3. It can be clearly seen that the objectives of speed and accuracy of the maneuver are indeed conflicting and thus each point on the Pareto front represents a compromise between them. The tail of the front (the right side, e.g. the points with high values of maximum deviation) corresponds to the solutions, in which almost no decrease of the average speed can be observed while the steering is performed exclusively by the control of the steering angle of the front wheels. Because of the high deviation from the reference trajectory these solutions are unacceptable for the robot control. In turn, the solutions with small value of maximum deviation are associated with significant deceleration and, as it can be seen from Fig. 3 (the plots, corresponding to 90 degrees turn), require significant change of all three control inputs. In general, we are interested in the most accurate performance possible. However, after some instant (left from the circle in Fig. 3) the average velocity of the maneuver drops dramatically without significant gain in accuracy. Thus we have selected that solution as a reasonable compromise at the point when further increase of performance would require significant drop of velocity. The performance of the selected solution is illustrated in Fig. 4. It must be noted that the produced solution is far non-trivial as it is associated with large slippage angle (the angle between the longitude axis of the robot and the instant linear velocity). It must be noted that the obtained trajectory is qualitatively similar to the one, obtained for human-controlled mobile robot, reported in Huang et al. (2005). More detailed discussion of the obtained solution is given below.

#### 4. CONTINUATION

At the second stage we want to obtain a continuous mapping from the desired turn angles to the parameters of the control inputs. We would like to note that the latter can be hardly achieved simply by applying to each turn angle the procedure, described in the previous paragraph. Indeed, since the described problem is expected to have numerous local extrema, such approach, applied for two close values of the turn angle, would lead to rather distant solutions. In order to avoid such behavior the solutions must be forced to stay close to each other, for example, by constraining the distance between the solutions in the space of parameters. Thus, here again we have conflicting

objectives: (i) to minimize the distance between the close solutions and at the same time to maximize (ii) average velocity of the maneuver and (iii) its accuracy.

This problem can be regarded in the framework of multi-objective optimization. In this case the Pareto front should be a surface in the three-dimensional space of the objective functions. However, solving the problem with 3 objective functions requires an order more computation than with 2. Thus, we decided not to optimize the average velocity of maneuver assuming that minimization of the distance between the solutions would in addition force the average velocities to be close to each other. From the Fig. 7A one can see that this assumption is reasonable enough.

To find the mapping between the turning angles and parameters of the control inputs, we start with solution selected for 90 degrees turn (Fig. 3). Then we compute the Pareto front for 95 degrees turn. The difference between the parameters is defined as following:

$$\rho(p_1, p_2) = \sqrt{\sum_{i=1}^{18} (p_{1i} - p_{2i})^2}$$

where vectors of parameters  $p_1$  and  $p_2$  were normalized in such a way that  $p_{ji} \in [0, 1]$ .

An example of the Pareto front for 95 degrees is provided in Fig. 5. It can be seen that the Pareto front has a flat tail (to the right from the circle) associated with nearly linear relationship between the difference from the previous solution and the accuracy of maneuver. The tail proceed to the right until it reaches the solution selected for the 90 degrees turn. The tail is almost flat, meaning that until some point significant gain in accuracy is achieved without deviating a lot from the previous solution. In turn, after that point dramatic growth of difference happens without bringing considerable increase in accuracy. Thus, it makes sense to select the corresponding solution as the best compromise between the two objectives.

The selected solution was used to compute the Pareto front for the 100 degrees turn and so on until 135 degrees. Similar procedure was performed in the direction of decrease of the turning angle until 50 degrees. Examples of Pareto fronts obtained this way for different turn angles are given in Fig. 5. It can be clearly seen that the described relationship between the objective functions holds for another turn angles as well.

Fig. 6 illustrates control inputs selected for different angles of turn. One can see that all three control functions are qualitatively the same for different angles of turn. In general, the maneuver starts with braking with both front and rear wheels, so that the linear velocity of the robot drops down (Fig. 7A). The rear wheels velocity reaches its minimum within about 0.5 seconds. After that a 2 seconds ‘‘plato’’ starts. During this time the rear wheels velocity increases just slightly. It must be noted that profile of the rear wheels velocity is quiet the same for all regarded angles. The front wheels velocity has nearly triangular form: first it decreases until some minimum and then increases with nearly the same magnitude of acceleration. The velocity profile is essentially the same for different turn angles, however it is shifted forward in time for sharper turns.

The steering angle has more complicated structure, which is significantly effected by the angle of turn. For all turn angles the steering starts with rather steep turn of the wheels, followed by the part with relatively slow change of the steering angle, after which it returns to zero rather steeply. For smaller angles of turn the initial steep part is shorter and has smaller amplitude than for larger. It ends approximately at the moment when the front wheels velocity reaches its minimum. The part of slow change is 1-1.5 seconds long and is associated with increase of the steering angle for the turn angles below 90 degrees and decrease otherwise. The start and the end of the whole steering process nearly coincide with those of the plato in the rear wheels velocity. On general, the controller commands appears to depend on the turn angle in rather continuous manner.

The examples of basic kinematic characteristics of the robot motion during maneuver are given in Fig. 7. One can see that the linear velocity of the robot has stereotypical profile, composed of constant deceleration, transients and acceleration. The first and the last part of the profile are nearly the same for all steering angles, while the transients differ in such manner that higher angle of turn is associated with lower average velocity. One can see that the decrease of front wheels angular velocity leads to almost unnoticeable change in the linear velocity of the robot. It happens because the tangential forces of the front and rear wheels nearly compensate each other. For easier understanding one might imagine a steady vehicles, in which the front and the rear wheels rotated in opposite directions with the same velocity.

The angular velocity is significantly more variable. For the smaller angles it seems that the robot slightly overturns and compensate it by negative value of the angular velocity at the end of maneuver. This compensation arises exclusively because of the properties of the wheel-road interaction, since no compensatory steering can be observed on Fig. 6. The slippage angle illustrates significant oversteering in the beginning of maneuver and slight understeering at the end.

## 5. CONCLUSION

In this paper we developed the approach to aggressive maneuver planning proposed in Velenis et al. (2007a,b) and inspired by the behavior of rally drivers. We slightly modified it in order to adapt to mobile robot application and gave an estimation of the Pareto front of the speed-accuracy relationship of the maneuver. The Pareto front estimate allowed us to propose a criteria for choosing the ‘‘best compromise’’ between those two objectives. After that we proposed a way to extend the solution for a range of different angles of the road turn. We show that the control inputs have rather stereotypical profiles, which depend continuously on the turn angle. The further development of this work can be performed in several direction. First of all, it seems reasonable to extend the current approach for different road conditions, i.e. different values of Coulomb friction coefficient. Next, it can be combined with existing algorithms of trajectory stabilization, like those proposed in Lucet et al. (2008b). And, finally, we would be interested in introducing feedback elements into the obtained control

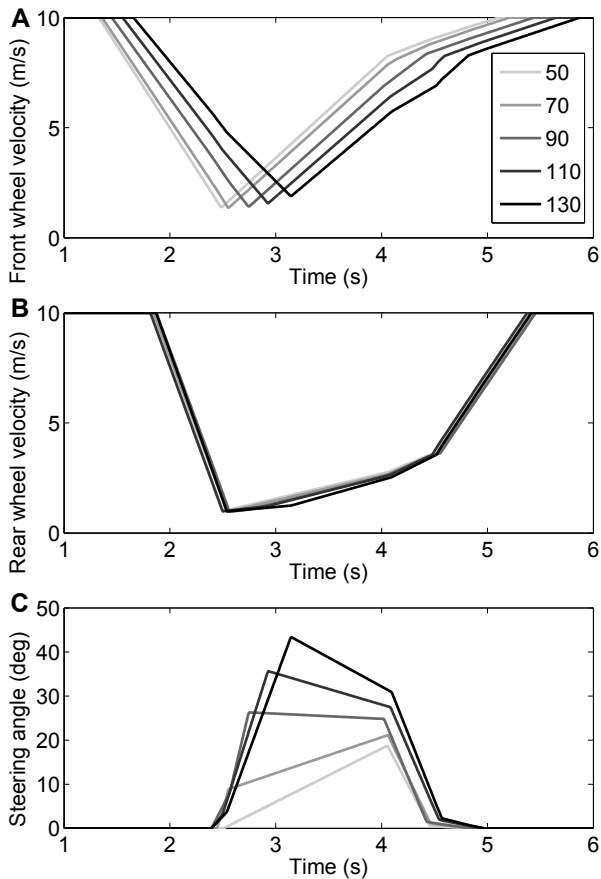


Fig. 6. Control parameters in solutions selected for different angles of the turn: *A* velocity of the front left wheel, *B* velocity of the rear left wheel, *C* steering angle of the front left wheel.

inputs, which are purely time-dependent at the moment. As the final goal of the current research we consider implementing the developed control algorithms on a mobile robot that is being currently designed at our institute.

#### REFERENCES

- Buehler, M., Iagnemma, K., and Singh, S. (2007). *The 2005 DARPA Grand Challenge - The Great Robot Race*, volume 36 of *Springer Tracts in Advanced Robotics*.
- Deb, K. (2001). *Multi-objectives optimization using evolutionary algorithms*. Wiley.
- Deb, K., Agrawal, S., Pratab, A., and Meyarivan, T. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lut-ton, J.J. Merelo, and H.P. Schwefel (eds.), *Proceedings of the Parallel Problem Solving from Nature VI Conference*, 849–858. Springer. Lecture Notes in Computer Science No. 1917, Paris, France.
- Huang, H., Chamberlain, L., and Murray, R. (2005). An experimental platform for motion estimation and maneuver characterization in high speed off-road driving. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 3090–3095.
- Lucet, E., Grand, C., Salle, D., and Bidaud, P. (2008a). Dynamic yaw and velocity control of the 6wd skid-steering mobile robot roburoc6 using sliding mode tech-

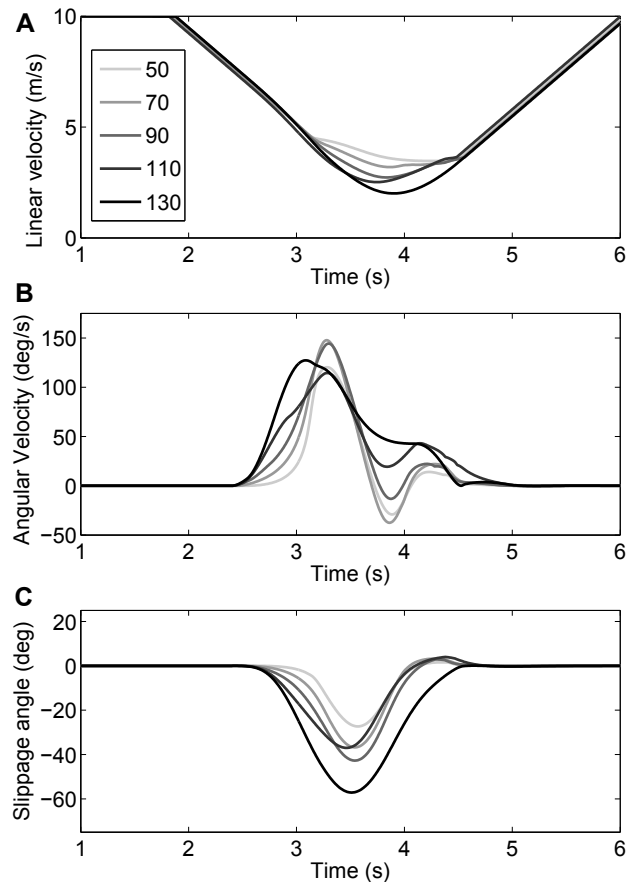


Fig. 7. Characteristics of robot maneuver in solutions selected for different angles of the turn: *A* linear velocity, *B* angular velocity, *C* slippage angle.

- nique. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 4220–4225.
- Lucet, E., Grand, C., Salle, D., and Bidaud, P. (2008b). Stabilization algorithm for a high speed car-like robot achieving steering maneuver. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2540–2545.
- Mouret, J.B. and Doncieux, S. (2010). Sferes\_v2: Evolving in the multi-core world. In *IEEE Congress on Evolutionary Computation, 2010 (CEC 2010)*.
- Pacejka, H. (2005). *Tire and Vehicle Dynamics*. SAE International, Elsevier, 2 edition.
- Peters, S. and Iagnemma, K. (2008). Mobile robot path tracking of aggressive maneuvers on sloped terrain. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 242–247.
- Velenis, E., Tsiotras, P., and Lu, J. (2007a). Aggressive maneuvers on loose surfaces: Data analysis and input parametrization. In *15th IEEE Mediterranean Control Conference, June 26-29, Athens, Greece*.
- Velenis, E., Tsiotras, P., and Lu, J. (2007b). Modeling aggressive maneuvers on loose surfaces: The cases of trail-braking and pendulum-turn. In *European Control Conference, Kos, Greece, July 2-5*.
- Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2), 173–195.