Strategy for automated dense parking: how to navigate in narrow lanes

Philip Polack, Louis-Marie Dallen, Aurélien Cord

Strategy for automated dense parking: how to navigate in narrow lanes*

Philip Polack[1], Louis-Marie Dallen[1] and Aurélien Cord[1]

***Abstract*— This paper presents the architecture of a high-density parking solution based on car-like robots specifically designed to move cars. The main difficulty is to park the vehicles close to one another which implies hard constraints on the robot motion and localization. In particular, this paper focuses on navigation in narrow lanes. We propose a Lyapunov-based control strategy that has been derived after expressing the problem in a Configuration Space formulation. The current solution has been implemented and tested on Stanley Robotics' robots and has been running in production for several months. Thanks to the Configuration Space formulation, we are able to guarantee the obstacles' integrity. Moreover, a method for calibrating the GPS orientation with a high-precision is derived from the present control strategy.**

I. INTRODUCTION

In recent years, the "parking difficulty" problem in modern cities has become more and more notable. Parking is one of the key links between the urban planning and transportation operation. The parking choice has a close relationship with holiday travel behavior [1].

Autonomous driving capabilities are of special interest considering the parking process as they relieve the user from the time-consuming task of finding a parking space. Furthermore, autonomous parking allows the vehicles to be arranged in a more dense geometric layout, as nobody needs to be able to access the car doors while they are parked. This increases the parking facility's capacity. Car makers pay much attention to automatic parking system and this driving technology has become one of the research hotspots [2], [3].

Therefore, Stanley Robotics proposes a new solution to address the parking problem: an automated parking management using robots. It comprises two main parts: at the higher-level, an intelligent parking management system (PMS) deals with the vehicle storage and the flow of vehicles. At the lower-level, a robot named Stan handles all the motions of the vehicles in the parking lot. The combination of these two parts increases the parking capacity significantly by densely storing the vehicles. However, this implies strong constraints on the robot motion in order to ensure the integrity of all the surrounding vehicles.

This paper deals with the navigation problem inside narrow lanes as shown in Figure 1 for our robot Stan. The focus is set on backwards driving in order to drop off or pick up a vehicle. A lane is composed of consecutive aligned parking spots. On each side of the lane, there are parking spots that may or may not be occupied as shown in Figure 2.

*This work was supported by the company Stanley Robotics

[1]Philip Polack, Louis-Marie Dallen and Aurélien Cord are with Stanley Robotics, 18 avenue Parmentier, 75011 Paris, France firstname.lastname@stanley-robotics.com

Fig. 1. Parking block with narrow lanes operated by Stanley Robotics.

As Stan is a car-like robot, it is subject to the slip-free rolling conditions which are non-holonomic constraints [4]. Due to its complexity, non-holonomic motion planning has been an attractive research field [5].

The motion planning of a car-like robot can be interpreted as a problem of finding a feasible path that connects the initial and the goal configurations while satisfying the non-holonomic constraints. Additional constraints are imposed due to obstacles [6], [7]. Non-holonomic and obstacle constraints are hard to consider simultaneously because conventional methods for the path planning of a car-like robot are not designed for dealing with obstacle regions [8]. Especially, in a narrow lane, where the robot's direction changes from forwards to backwards and vice versa, obstacle constraints cause major difficulties in path planning [9], [10].

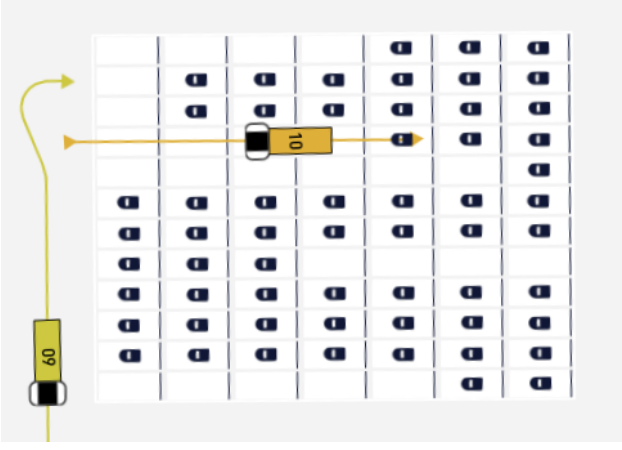

Fig. 2. Organisation of the parking spots. The black cars represent occupied parking spots (representation not at scale). Two robots and their associated path are shown in green and orange: the robot #09 (green) is in normal navigation mode while the number #10 (orange) is in narrow-lane navigation mode.

In our solution, the motion planning problem is composed of two steps. At the high-level, the PMS provides to the robot a collision-free path to the parking spot of the vehicle. In particular, the parking spaces are composed of long straight line corridors with a given width. However, the PMS does not provide how to navigate in narrow lanes without collision. Real-world scenarios require high-precision controllers that can guarantee the collision-free motion. In particular, they need to be robust to modeling errors of the path planner and external disturbances. In [11], an adaptive sliding controller robust to disturbances and model uncertainty is presented for a quadrotor. [12] introduces a Lyapunov stable control-law for parking of a car-like robot that guarantees global limits of the system under bounded state estimation uncertainties.

In our case, at the low-level, the robot is composed of a Lyapunov-based control-law which stabilizes the system on the path. The Lyapunov control law also ensures the integrity of the robot by preventing from overshooting the lane limits. One of the novelties of our approach is the reformulation of the problem in the Configuration Space (*CS*) which enables it to predict the future position of the robot.

This paper is organized as following. There are two aspects of the vehicle that have to be taken into account during the obstacle avoidance process: shape and kinematics. The shape is a geometric problem that we tackle in section II by expressing the problem in the Configuration Space. The kinematics involves choosing a safe and stable control-law that is described in section III. Section IV presents the implementation of this control strategy on our robots. One interesting application is that it provides a simple and cheap method for high-precision GPS calibration. At last, section V concludes this paper.

II. FORMULATION OF THE PROBLEM

In order to achieve a high-density parking, the architecture of our system is organized in two main layers. At the higher-level, the PMS sends a path to the robot in order to get to the right parking spot as shown for the two robots in Figure 2. Indeed, the PMS knows all the available parking spots and how to reach them. The parking spots are organized into blocks of several lanes. Figure 2 represents one of these blocks composed of free and occupied parking spaces. At the lower-level, the robot follows the path thanks to two different control-laws. On the one hand, moving around between blocks is a classic control problem that has been studied widely in the literature: see for example [13] and references therein. On the other hand, moving inside the blocks is a challenging problem since the lane widths are small with respect to the size of the robot. Thus, it is necessary to design a high-accuracy controller that maintains the robot inside its lane at all time. Inside a block, the PMS provides a straight path corresponding to the middle of the lane as shown in orange in Figure 2. This path is feasible as long as the width of the robot, denoted by w_{robot}[1] is strictly smaller than the width of the lane w_{lane} minus the localization errors.

[1]If the vehicle is larger than the robot, we consider the robot's size to be the union of the vehicle and the robot.

Designing a control law that prevents the robot from overshooting the lane limits is a complicated task due to on the one hand the geometric shape of the robot and on the other hand the non-holonomic constraints. While the primer makes it complicated to check whether a given position of the robot collides with the lane limits, the second restricts the admissible motion of the car-like robot.

A. Formulating the problem in the Configuration Space

Studying the motion of a two or three-dimensional object is a complicated task due to its geometry, in particular to check for collision-avoidance. This has lead to the famous piano-mover problem, introduced in [14]: given a map of the environment with obstacles, and an initial and a goal configurations of an object, is it possible to find an admissible collision-free path between these two configurations? In [15], the author has shown that any admissible motion of a three-dimensional mechanical system appears as a collision-free path for a point in the Configuration Space (*CS*), i.e. the space of all the possible configurations of the object. This greatly simplifies the problem as it transforms the problem into searching if the initial and goal configurations belongs to the same connected component.

In the case of narrow lane navigation, the robot footprint is entirely defined by three configurations: its longitudinal error $\tilde{x}$, its lateral error $\tilde{y}$ and its angular error $\tilde{\theta}$ as shown in Figure 3. These configurations are expressed at the control point which is the middle of the rear axle (represented by a red dot on the figure). As the robot has to remain in its lane, the two adjacent lanes are considered as obstacles and represented in gray on the figure. Moreover, due to the symmetry of the problem along the $\tilde{x}$-axis, in order to check if a configuration is collision-free, we can focus on the projection of the Configuration Space on the $(\tilde{\theta}, \tilde{y})$ plane. The $\tilde{x}$ will be only necessary to check whether the goal, *i.e.* the parking spot of the vehicle, has been reached.

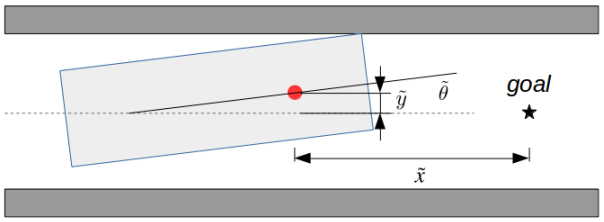

Fig. 3. The three configurations $(\tilde{x}, \tilde{y}, \tilde{\theta})$ that determine the robot's footprint (light gray). The dark gray rectangles represent the beginning of the adjacent lanes

In order to determine CS_{free}, the set of all the configurations in *CS* that are collision-free, we compute its limits corresponding to the case where the robot is in contact with one of the adjacent lanes. This leads to Equations (1-4) where fr, fl, rr and rl correspond respectively to the cases where the front-right, front-left, rear-right and rear-left

corner of the robot touches one of the lane limits:

$$\tilde{y}_{fr} = \frac{w_{lane}}{2} - A\sin(\tilde{\theta}_{fr}+\alpha) \quad (1)$$

$$\tilde{y}_{fl} = A\sin(\tilde{\theta}_{fl}+\alpha) - \frac{w_{lane}}{2} \quad (2)$$

$$\tilde{y}_{rr} = \frac{w_{lane}}{2} - B\sin(\tilde{\theta}_{rr}+\beta) \quad (3)$$

$$\tilde{y}_{rl} = B\sin(\tilde{\theta}_{rl}+\beta) - \frac{w_{lane}}{2} \quad (4)$$

where A, B, α and β are given by Equations (5-8) and depend on some dimensions of the robot (see Figure 4): its wheelbase L_{wb}, its width w_{robot} and the distance between the middle of the front (resp. rear) axle and the front (resp. rear) of the robot p_{fr} (resp. p_{rr}).

$$A = \sqrt{\left(\frac{w_{robot}}{2}\right)^2 + (L_{wb}+p_{fr})^2} \quad (5)$$

$$B = \sqrt{\left(\frac{w_{robot}}{2}\right)^2 + p_{rr}^2} \quad (6)$$

$$\alpha = \arctan\left(\frac{w_{robot}}{2(L_{wb}+p_{fr})}\right) \quad (7)$$

$$\beta = \arctan\left(\frac{w_{robot}}{2p_{rr}}\right) \quad (8)$$

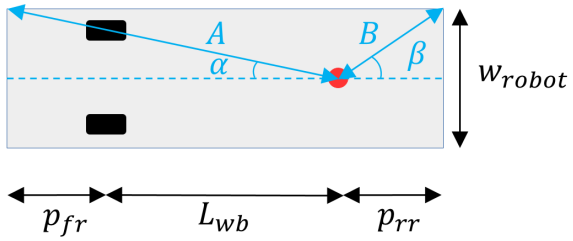

Fig. 4. General robot description.

In the case considered, as w_{lane} is small compared to the dimensions of the robot, $\tilde{\theta}$ remains small (typically less than 3°). Thus Equations (1-4) can be approximated by affine functions given by Equations (9-12) as $A\sin\alpha = B\sin\beta = w_{robot}/2$ and $A\cos\alpha = L_{wb}+p_{fr}$ and $B\cos\beta = p_{rr}$. This approximation error is negligible compared to the localization noise and greatly simplifies the shape of CS_{free}:

$$\tilde{y}_{fr} = \left(\frac{w_{lane}-w_{robot}}{2}\right) - (L_{wb}+p_{fr})\tilde{\theta}_{fr} \quad (9)$$

$$\tilde{y}_{fl} = \left(\frac{w_{robot}-w_{lane}}{2}\right) - (L_{wb}+p_{fr})\tilde{\theta}_{fl} \quad (10)$$

$$\tilde{y}_{rr} = \left(\frac{w_{lane}-w_{robot}}{2}\right) + p_{rr}\tilde{\theta}_{rr} \quad (11)$$

$$\tilde{y}_{rl} = \left(\frac{w_{robot}-w_{lane}}{2}\right) + p_{rr}\tilde{\theta}_{rl} \quad (12)$$

Thus CS_{free} is described by Equation (13) and shown in Figure 5. The configuration $(0, \frac{w_{lane}-w_{robot}}{2})$ called $C1$ in Figure 5 corresponds to the case where the robot is at contact on the whole right side as shown in Figure 6; the configuration $\left(\frac{w_{lane}-w_{robot}}{L_{wb}+p_{fr}+p_{rr}}, \frac{w_{lane}-w_{robot}}{2}\left(1-\frac{2p_{rr}}{L_{wb}+p_{fr}+p_{rr}}\right)\right)$ called $C2$ corresponds to the situation where the robot is at contact with the obstacles at both the rear-right and front-left corners as shown in Figure 7.

$$CS_{free} = \quad (13)$$
$$\{(\tilde{\theta},\tilde{y}) \text{ s.t. } \tilde{y}_{fl}(\tilde{\theta}) \le \tilde{y} \le \tilde{y}_{fr}(\tilde{\theta}),\ \tilde{y}_{rl}(\tilde{\theta}) \le \tilde{y} \le \tilde{y}_{rr}(\tilde{\theta})\}$$

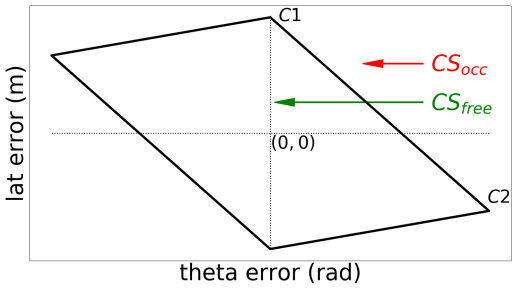

Fig. 5. Collision-free Configuration Space CS_{free} and occupied Configuration Space CS_{occ}.

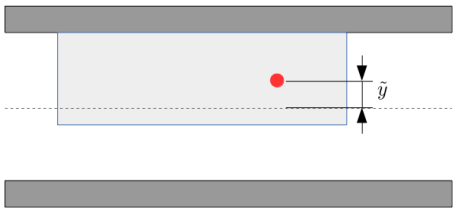

Fig. 6. Configuration $C1$: the right edge of the robot (light gray) touches the limit of the lane. The gray rectangles represent the beginning of the adjacent lanes

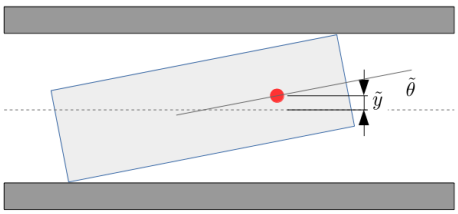

Fig. 7. Configuration $C2$: two opposite corners of the robot (light gray) touch the limits of the lane. The gray rectangles represent the beginning of the adjacent lanes

B. Handling the non-holonomic constraints

A non-holonomic constraint for a system is a constraint that does not reduce the dimension of its Conguration Space: given an initial configuration, the robot can still reach any other configuration. In other words, it is a non-integrable constraint. In the case of a car-like robot, the slip-free rolling condition is a non-holonomic constraint. However this constraint restricts the admissible motions: for example, a

car-like robot cannot reduce its lateral error while keeping its angular error at zero.

This has a strong impact on the design of a control law as any collision-free path in the configuration space does not necessarily correspond to an admissible collision-free path in the physical space. In particular, one interesting observation is that the closer the angular error is to zero, the longer the lateral error takes to converge.

In order to take into account the non-holonomic constraints, we designed the control-law based on a kinematic bicycle model [16]. This model is a good approximation of the robot behavior as we are only considering low speed applications where dynamics effects can be neglected. As the reference heading θ_r remains constant, considering a frame where the x-axis is aligned with the central line of the lane, the kinematic bicycle model can be expressed as in Equations (14-16) where V_r is the robot speed on the rear axle and δ_{fr} the front steering angle.

$$\dot{\tilde{x}} = V_r \cos\tilde{\theta} \quad (14)$$

$$\dot{\tilde{y}} = V_r \sin\tilde{\theta} \quad (15)$$

$$\dot{\tilde{\theta}} = \frac{V_r}{L_{wb}} \tan\delta_{fr} \quad (16)$$

This model takes into account the slip-free rolling condition given by Equation (17):

$$\tilde{x}\sin\tilde{\theta} - \tilde{y}\cos\tilde{\theta} = 0 \quad (17)$$

III. CONTROL STRATEGY FOR NARROW LANE NAVIGATION

Once the problem has been reformulated in the Configuration Space, it becomes easier to design a control law that is guaranteed to be collision-free. Starting from any configuration $(\tilde{\theta}_0, \tilde{y}_0)$ in the free-configuration space, we need to design a response that converges to $(0,0)$ without exiting the collision-free configuration space CS_{free} while respecting the kinematic bicycle model.

A. Lyapunov Stability

For that purpose, we used a control-law based on Lyapunov stability as in [17] where a control strategy to move a car-like robot to a desired posture within prescribed boundaries is presented. Assuming that the robot has a constant speed V_r, the control-law implemented on our robots is given by Equation (18).

$$\delta_{fr} = \arctan\left(-L_{wb}\left(k_{ang}\,\mathrm{sign}(V_r)\tilde{\theta} + k_{lat}\,\mathrm{sinc}(\tilde{\theta})\tilde{y}\right)\right) \quad (18)$$

Consider the function V given by Equation (19). V is a Lyapunov function associated to the system given by Equations (14-16).

$$V(\tilde{\theta},\tilde{y}) = \frac{1}{2}\left(k_{lat}\tilde{y}^2 + \tilde{\theta}^2\right) \quad (19)$$

Proof:

- $V(0,0) = 0$
- $\forall(\tilde{\theta},\tilde{y}) \in \mathbb{R}^2, (\tilde{\theta},\tilde{y}) \neq (0,0) \Rightarrow V(\tilde{\theta},\tilde{y}) \geq 0$
- Let's proove that $\forall(\tilde{\theta},\tilde{y}) \in \mathbb{R}^2, (\tilde{\theta},\tilde{y}) \neq (0,0) \Rightarrow \dot{V}(\tilde{\theta},\tilde{y}) \leq 0$

$$\dot{V} = k_{lat}\tilde{y}\dot{\tilde{y}} + \tilde{\theta}\dot{\tilde{\theta}} \quad (20)$$

$$= V_r\left(k_{lat}\tilde{y}\sin\tilde{\theta} + \frac{\tilde{\theta}}{L_{wb}}\tan\delta_{fr}\right) \quad (21)$$

Choosing δ_{fr} such as in Equation (18), one obtains that V is a Lyapunov-function as (we assume that V_r is constant during the motion):

$$\dot{V} = -k_{ang}\tilde{\theta}^2 < 0 \quad (22)$$

■

Thus, for all pair $(k_{ang}, k_{lat}) \in {\mathbb{R}_+^*}^2$, the system is stable for all $(\tilde{\theta},\tilde{y}) \in \mathbb{R}^2$ using the control-law given by Equation (18) and converges to zero. However, in order to ensure that the robot stays inside CS_{free}, it is necessarily to tune the gains k_{lat} and k_{ang} correctly.

Remark 1: If we consider different gains for forwards and backwards driving, we end up with a switched system where each subsystem is Lyapunov-stable. However, the global system has no guarantee to be stable as explained in [18]. This problem can be overcome by choosing the same value of k_{lat} in both directions as the function V given by Equation (19) is a Lyapunov function for the whole system in that case.

B. Theoretical Tuning of the Gains

The tuning of the control gains k_{lat} and k_{ang} is a complicated but important task. The choice is a compromise between many criteria such as noise sensitivity, convergence speed and overshoots. In our case, it becomes also critical for the safety of the robot and its environment.

Backwards driving is a more challenging task than forwards driving due to the lever arm of the robot as the control point is located close to the rear of the robot. Thus, we focus this paper on the response in the CS to different gains k_{lat} and k_{ang} as shown in Figures 8 to 11 for backwards driving. More precisely, a grid of initial configurations in CS_{free} is generated; then for each initial configuration, the evolution of the path during the next 3m is simulated using the kinematic bicycle model. It is interesting to notice that whatever the gains are, many initial configurations belonging to CS_{free} lead to a path that does not remain inside CS_{free}. This becomes the case for almost all initial configurations in CS_{free} as the value of k_{lat} increases towards the value of k_{ang} (see Figures 10 and 11).

For a better understanding, the iso-steering levels have been added on each figure. They correspond to the case where δ_{fr} is constant. Considering Equation (18) and only first order terms in $\tilde{\theta}$ (as $\tilde{\theta} << 1$), the different iso-steering levels are given by the following affine function:

$$\tilde{y} = -\frac{k_{ang}}{k_{lat}}\mathrm{sign}(V_r)\tilde{\theta} - \frac{\tan\delta_{fr}}{k_{lat}L_{wb}} \quad (23)$$

Hence, for backwards driving, the slope of the iso-steering 0 is given by the ratio k_{ang}/k_{lat}. We observe that the paths in the configuration space all tends toward this line, meaning

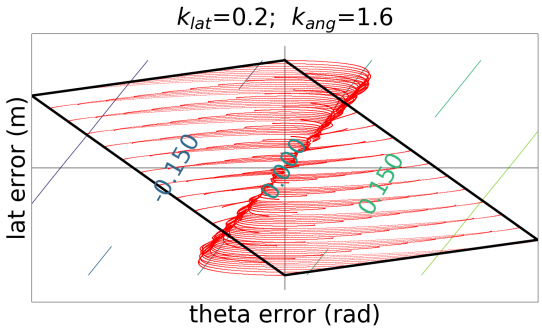

Fig. 8. Theoretical response of the control-law for backwards driving with $k_{lat} = 0.2$ and $k_{ang} = 1.6$.

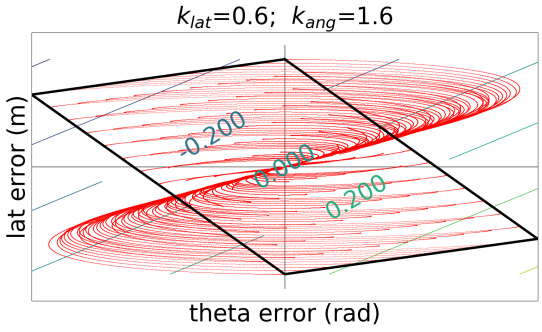

Fig. 9. Theoretical response of the control-law for backwards driving with $k_{lat} = 0.6$ and $k_{ang} = 1.6$.

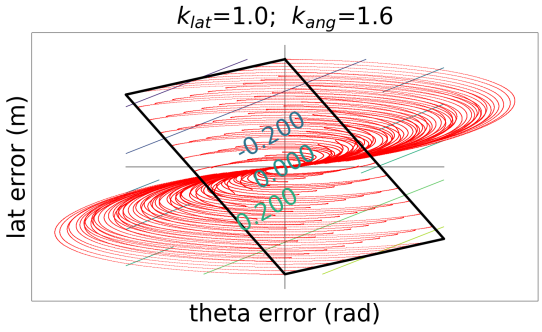

Fig. 10. Theoretical response of the control-law for backwards driving with $k_{lat} = 1.0$ and $k_{ang} = 1.6$.

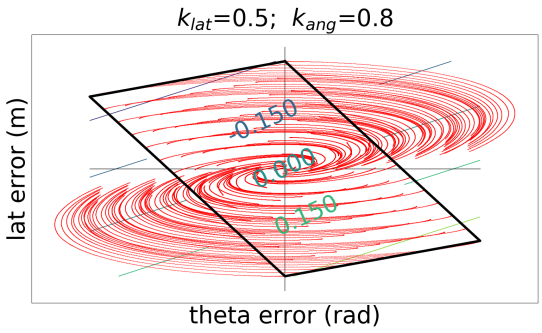

Fig. 11. Theoretical response of the control-law for backwards driving with $k_{lat} = 0.5$ and $k_{ang} = 0.8$.

that the robot finishes converging with almost no steering. Thus, the response depends mostly on the ratio k_{ang}/k_{lat}. The higher it is, the more initial configurations will remain inside CS_{free}. However, higher ratio also leads to a slower response as the system will correct the errors with a smaller δ_{fr} value. Thus, as the length of the lane is not infinite, it is important to choose a good trade-off between these two properties.

At last, Figures 10 and 11 show the response of the system with two pairs of gains having the same ratio. The response are quite similar but the convergence is slower in the cases where the gain are smaller as the paths converge faster to the zero error configuration.

As mentioned earlier, it is not possible to tune the gains for backwards driving in order to converge to zero from all configurations in CS_{free} without collision. Thus, it is critical to check before the robots enter a lane whether the initial configuration leads to a collision-free path or not. In the case the path does not remain inside CS_{free}, the robot needs to maneuver outside of the lanes, where there is no obstacles. This maneuver is out of the scope of the present paper. However, choosing a good tuning enable to maximize the chance that the initial configuration of the robot leads to a path that remains inside CS_{free}: for that purpose, the higher the ratio k_{ang}/k_{lat} is, the better it is. However, this is at a cost of a slower convergence rate. A good tuning is thus a compromise between these two criteria.

IV. IMPLEMENTATION ON ROBOT

The control strategy presented in the previous section has been applied on the robot Stan. The localization is based on the fusion between a 50Hz dual antennas Real-Time Kinematic GPS, a SLAM and odometry. The control point is located at the center of the rear axle.

A. Initial experimental trials

First, we run some experiments where the robot had to follow a straight line outside of a lane, starting with an initial lateral and angular error $(\tilde{\theta}_{init}, \tilde{y}_{init})$. The measurements obtained are shown in blue dots on Figure 12. Then, we compared these results with the theoretical ones, which are displayed in cyan: the two curves do not match as the experimental results converge to a configuration $(\tilde{\theta}_{\infty}, \tilde{y}_{\infty})$ that is not $(0,0)$ which should not be possible in theory. Repeating the same experiment starting from a different initial configuration, we observed that the path converges again to the same final configuration.

Going further in the analysis, it turns out that this configuration lies on the curve $\delta_{fr} = 0.0$. This means that from this configuration $(\tilde{\theta}_{\infty}, \tilde{y}_{\infty})$, the robot should converge to zero "naturally", without any steering angle applied. Thus, this proves a misalignment between the GPS and the wheels of the robot of $\tilde{\theta}_{\infty}$. Hence, we added in Figure 12 the theoretical results obtained by fixing this angular offset, simply by changing $\tilde{\theta}$ by $\tilde{\theta} + \tilde{\theta}_{\infty}$ in the control-law: the results are shown in orange. This angular offset was very small on each of our robots (less than 1°) but sufficient

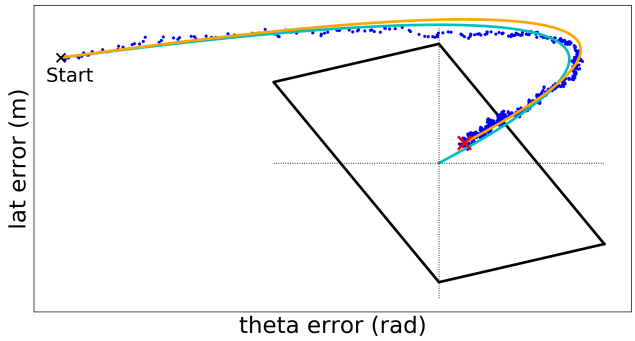

Fig. 12. Comparing experimental results (blue dots) with the theory before (cyan) and after (orange) GPS offset was added. The red cross indicates the configuration $(\tilde{\theta}_\infty, \tilde{y}_\infty)$ where the experimental results converge to.

to decrease the performance and the safety of the control-law in the narrow lanes given the small margins available. Thus, this initial experiment enabled us to correct the misalignment between the GPS and the wheel with an easy and accurate method, hence being robust to industrial assembly imperfections.

B. Results after GPS calibration

After aligning the GPS on the wheel orientation, the experiments were run again. Starting from an initial configuration $(\tilde{\theta}_{init}, \tilde{y}_{init})$, Figure 13 shows that the experimental results (in blue dots) match perfectly the theoretical ones (in cyan) and converge to zero. Only a small overshoot of the experimental results in the top right dial can be observed and is caused by an unmodeled actuator delay.

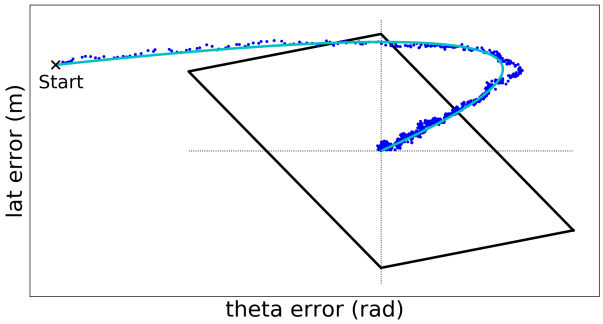

Fig. 13. Comparing experimental and theoretical results after GPS offset was added.

The presented control strategy has thus been implemented on all the robots operated by Stanley Robotics. The tuning of the control gains is the same on all robots but the GPS calibration changes. Before entering a narrow lane, we check whether the configuration will remain inside CS_{free}: this is done using the simulation model given by Equations (14-16) to which we add some measurement noise. This noise is a Gaussian noise that has the same characteristic as our localization module. After running several months in production, no major issue has been reported. A video of a robot moving inside a narrow lane is available at the following link: shorturl.at/cjwGW

C. Comparison with a pure-pursuit controller

Figure 14 compares the experimental results for the present Lyapunov control method (in blue dots) with a pure-pursuit controller [19] (in red dots) starting from the same initial configuration. The theoretical results for the Lyapunov controller have been added in cyan. We observe not only that the Lyapunov method is safer as it remains better confined in CS_{free}, but also that the pure-pursuit does not converge to $(0,0)$ accurately. This has been confirmed by several other experiments and can be explained by the fact that the pure-pursuit controller draws a circular arc from the rear axle of the robot to a look-ahead point. In our case, as the path is a straight line, the circular arc is "degenerated".

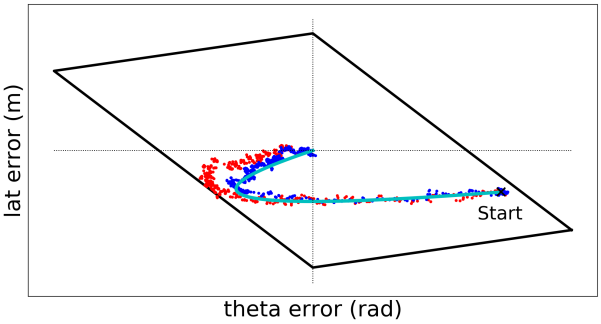

Fig. 14. Comparison between the presented Lyapunov-based controller law (in blue) and a pure-pursuit controller (in red). The theoretical results using the Lyapunov-based control law are represented in cyan.

V. CONCLUSIONS

This paper presented a control strategy adapted to narrow lane navigation. By reformulating the problem in the Configuration Space, we were able to design a Lyapunov-stable control-law that guarantees the integrity of the robot and its environment at all time. By considering the adjacent lanes as obstacles, the robot is able to handle not only static environments but also dynamic ones if another robot operates in the adjacent lane. The obstacle integrity is achieved by predicting the future configurations of the robot using a kinematic bicycle model. Thus, it is possible to predict at the entrance of the lane whether the robot will remain inside the lane or not. These theoretical results have then been validated in practice on real robots. Moreover, a simple, cheap and highly-accurate method for calibrating the GPS heading with the zero of the wheels is proposed.

Future works will be focusing on one hand on the maneuver strategy to enter a lane whatever the initial configuration is and on the other hand on an automatic gain tuning method using optimal control. Deriving a procedure for automatic tuning of the GPS offset would also be of high interest.

ACKNOWLEDGMENT

The authors would like to thank the operation and robot software teams of Stanley Robotics for their support and feedbacks in order to implement and test the algorithms, and the marketing team for providing the pictures and the video.

REFERENCES

[1] B. Wang, C. Shao, J. Li, D. Zhao, and M. Meng, "Investigating the interaction between the parking choice and holiday travel behavior," *Advances in Mechanical Engineering*, vol. 7, no. 6, p. 1687814015589499, 2015.

[2] W. Wang, Y. Song, J. Zhang, and H. Deng, "Automatic parking of vehicles: A review of literatures," *International Journal of Automotive Technology*, vol. 15, no. 6, pp. 967–978, 2014.

[3] S. Klemm, M. Essinger, J. Oberländer, M. R. Zofka, F. Kuhnt, M. Weber, R. Kohlhaas, A. Kohs, A. Roennau, T. Schamm, *et al.*, "Autonomous multi-story navigation for valet parking," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1126–1133, IEEE, 2016.

[4] J.-P. Laumond, "Feasible trajectories for mobile robots with kinematic and environment constraints," in *Proc. International Conference on Intelligent Autonomous Systems*, pp. 346–354, 1986.

[5] Z. Li and J. Canny, *Nonholonomic Motion Planning*. Springer Science & Business Media, 1993.

[6] J. Minguez, F. Lamiraux, and J.-P. Laumond, "Motion planning and obstacle avoidance," in *Springer handbook of robotics*, pp. 1177–1202, Springer, 2016.

[7] Y. Hua, H. Jiang, Y. Cai, X. Zhang, S. Ma, and D. Wu, "Path tracking control of automatic parking cloud model considering the influence of time delay," *Mathematical Problems in Engineering*, vol. 2017, 2017.

[8] W. Liu, Z. Li, L. Li, and F.-Y. Wang, "Parking like a human: A direct trajectory planning solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3388–3397, 2017.

[9] D. Kim and W. Chung, "Motion planning for car-parking using the slice projection technique," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1050–1055, IEEE, 2008.

[10] D. Xu, Y. Shi, and Z. Ji, "Model-free adaptive discrete-time integral sliding-mode-constrained-control for autonomous 4wmv parking systems," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 834–843, 2017.

[11] B. T. Lopez, J.-J. Slotine, and J. P. How, "Robust collision avoidance via sliding control," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2962–2969, IEEE, 2018.

[12] A. Tayebi and A. Rachid, "A time-varying-based robust control for the parking problem of a wheeled mobile robot," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3099–3104, IEEE, 1996.

[13] 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," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[14] J. T. Schwartz and M. Sharir, "On the "piano movers'" problem i. the case of a two-dimensional rigid polygonal body moving amidst polygonal barriers," *Communications on pure and applied mathematics*, vol. 36, no. 3, pp. 345–398, 1983.

[15] Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE Transactions on Computers*, vol. C-32, pp. 108–120, Feb 1983.

[16] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[17] S. Lee, M. Kim, Y. Youm, and W. Chung, "Control of a car-like mobile robot for parking problem," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 1, pp. 1–6, IEEE, 1999.

[18] M. S. Branicky, "Multiple lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Transactions on automatic control*, vol. 43, no. 4, pp. 475–482, 1998.

[19] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," tech. rep., Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, 1992.