Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 9282–9287

Stability focused evaluation and tuning of special ground vehicle tracking algorithms

Peter Bauer [*,**] **Mihaly Nagy** [*,**] **Gergely Istvan Kuna** [*,**] **Adam Kisari** [*,**] **Erno Simonyi** [*,**] **Antal Hiba** [*,***] **Istvan Drotar** [****]

[*] *Research Center of Vehicle Industry, Széchenyi István University, Győr, Hungary*

[**] *Systems and Control Laboratory, Institute for Computer Science and Control, ELKH, Budapest, Hungary (e-mail: bauer.peter@sztaki.hu)*

[***] *Computational Optical Sensing and Processing Laboratory, Institute for Computer Science and Control, ELKH, Budapest, Hungary*

[****] *Széchenyi István University, Győr, Hungary*

Abstract: This paper deals with a special tracking problem when a ground vehicle should be tracked by a multicopter flying ahead of the vehicle. Pre-designed vehicle route is assumed and the UAV stops or slows down at every intersection to react to route changes. After introducing the problem, the methods applied in a real flight demonstration in the Smart City module of ZalaZONE proving ground are presented. Then new methods are introduced to possibly improve performance. The main focus of the article is the evaluation of the stability of the methods and the provision of tuning guidelines. All of the introduced methods is tuned based-on the guidelines considering real ground vehicle test data and the high fidelity simulation of the applied multicopter. The two best methods are compared in detail and guidelines of their applicability are provided.

Keywords: vehicle tracking, forerunner UAV, stability, velocity scaling control, PD control, freedrive, routedrive

1. INTRODUCTION

The forerunner UAV concept was introduced in Nagy (2021). It means a camera equipped UAV flying in front of and above emergency ground vehicles (EGVs) to notify the driver about the hidden threats covered by buildings, vegetation or other vehicles. A real life demonstration of the whole system was presented in fall 2022 (Vehicle Industry Research Center (2022)). A main challenge of this concept is to fly ahead of the EGV but follow its route and the sudden route changes (decided by the driver based-on his/her experience). There is a wide range of literature about tracking ground vehicles with rotary or even fixed-wing aerial vehicles e. g. Ariyur and Fregene (2008), Greatwood et al. (2017), Xiao et al. (2018) and Hentati and Fourati (2020) the latter giving an overview about the methods. However, no such special case is discussed. Nagy (2021) introduces the freedrive concept when the drone flies to the next intersection, waits for the ground vehicle and then moves forward. The routedrive concept was introduced in Hiba et al. (2022) flying only braking distance ahead of the EGV and 'pulling back' the drone above it near intersections. This guarantees more continuous motion of the drone without stopping except for the case when the EGV stops. For the fall 2022 demonstration the routedrive concept was tuned (in simulation and real flight) applying velocity scaling (VS) or PD control solutions for drone velocity. Based-on in-flight tuning finally the PD solution was applied (see Fig. 3). After summarizing the routedrive concept and its real flight test results and revisiting the freedrive concept the current article focuses on the stability analysis of the solutions and the related tuning guidelines. Exhaustive tuning of the methods is done based-on the developed guidelines and finally, application suggestions are given.

Section 2 summarizes the routedrive methods and the flight test results. Section 3 revisits the freedrive concept in the same framework. Section 4 makes detailed evaluation of the stability of the methods. Section 5 summarizes the tuning results and makes application suggestions. Finally, Section 6 concludes the paper.

2. ZALAZONE DEMONSTRATION CONTROLLERS AND RESULTS

In the forerunner concept it is assumed that the EGV route is designed in advance and sent to the drone upon system startup. In case of sudden route change by the EGV driver the re-designed route is sent again. Known route parameters are assumed to be intersection center position C_i, width W_i and length L_i as shown in Fig. 1. A restriction in the current development was to assume intersections of four roads with rectangular area. This can be relaxed later.

Other intersection parameters are χ_{IN} direction of the road on which the EGV enters, IN_i position of enter line center, O_{1i} to O_{3i} positions of exit line centers. Enter and

10.1016/j.ifacol.2023.10.212

exit lines are parallel to the roads. Note that in case of the start intersection C_0 the χ_{IN} direction is calculated between car position (P_c) and center while otherwise it is determined by the previous C_{i-1} and current C_i centers (see Fig. 1). The figure also shows that outside the intersection the X direction is along the road and Y is perpendicular to it while inside X is aligned with multicopter heading and Y is perpendicular to it. From now on the perpendicular direction is called cross distance while the other is the along distance.

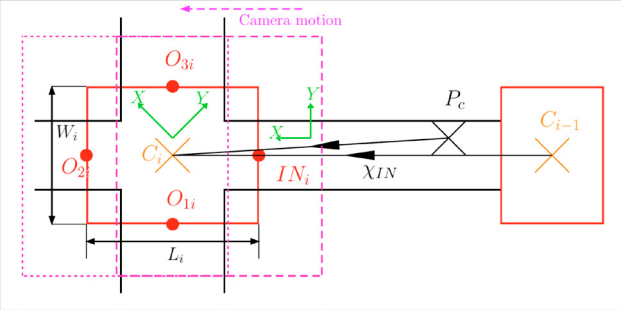

Fig. 1. The parameters of intersections

To define the tracking methods the positions of drone (P_d), car (P_c) and intersection (P_{IN_i}) along the road (X direction) and additional parameters and functions should be considered as shown in Fig. 2.

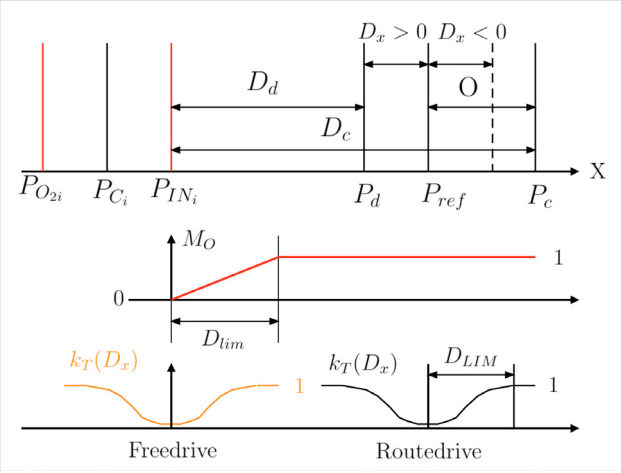

Fig. 2. Tracking parameters along the road

In the ZalaZONE demonstration the routedrive concept introduced in Hiba et al. (2022) was applied with velocity scaling (VS) and PD control solutions. In this, outside the intersection the drone should fly braking distance O_x ahead of the EGV but when approaching the intersection it is 'pulled back' above the vehicle with the $M_O = \frac{D_d}{D_{lim}}$ ($D_d \leq D_{lim}$) linear scaling starting D_{lim} distance before the intersection (see Fig. 2). Inside the intersection the drone should be above the EGV and follow it including its heading. The measured braking distance of a Skoda Rapid vehicle $O_x = 4 + 0.1\|V_c\|^2$ is considered here with $\|V_c\|$ absolute speed.

In the VS approach a distance dependent velocity scaling function (1) is applied to slow down the drone when it approaches the reference position (P_{ref}) (visualized in the bottom of Fig. 2). In the PD control a proportional gain g_P is applied to the position and a derivative g_D to the velocity error. The two control approaches are summarized in Table 1.

$$k_T(D) = \frac{sign(D)}{2}\left(1 - cos\left(\frac{D \cdot \pi}{D_{LIM}}\right)\right), \ |D| \leq D_{LIM}$$
$$k_T(D) = sign(D) \quad \text{otherwise} \tag{1}$$

Table 1. Summary of EGV tracking concepts and methods (given formula valid for both concepts if no specific concept is noted)

Velocity error	$\Delta V_j = V_{cj} - V_{dj}$
Outside the intersection	
Routedrive position error	$D_j = D_{cj} - M_{Oj}O_x - D_{dj}$
Freedrive position error	$D_j = P_{INj} - P_{dj}$
Routedrive VS reference	$V_{jref} = V_{cj} + k_T(D_j)Vmax_j$
Freedrive VS reference	$V_{jref} = k_T(D_j)Vmax_j$
Routedrive PD reference	$V_{jref} = V_{cj} + g_P D_j + g_D \Delta V_j$
Freedrive PD reference	$V_{jref} = g_P D_j$
Inside the intersection	
Position error	$D_j = P_{cj} - P_{dj}$
VS reference	$V_{jref} = V_{cj} + k_T(D_j)Vmax_j$
PD reference	$V_{jref} = V_{cj} + g_P D_j + g_D \Delta V_j$

Note that the basic velocity reference is the car velocity and additional terms are applied to compensate the position and velocity tracking error. $j \in [x, y]$ shows the direction of the control (along or cross) and $M_{Oy} = 0$ meaning no offset in the cross direction. The tunable control parameters are $Vmax$, D_{LIM}, D_{lim}, g_P and g_D. It is practical to define $Vmax$ and D_{LIM} dependent on the absolute car velocity: $Vmax_j = ssc_j \cdot \max([k_V\|V_c\|, 2])$, $D_{LIM} = k_D\|V_c\|$. $ssc_x = 1$ and $ssc_y \leq 1$ to make cross distance tracking less agile. Note that the minimum $Vmax$ value was set to be $2m/s$ to be able to compensate disturbances even with stopped EGV.

The real flight demonstration drone was the DJI M600 Pro hexacopter. Its high fidelity simulation model (in-flight identified in the project) was applied for pre-tuning of the controllers. Then in-flight trial and error tuning was done in ZalaZONE Smart City tracking a car on a given route (see Fig. 3). Besides trial and error systematic tuning (e. g. Ziegler-Nichols) methods could be applied but system construction did not allow on-line change of the parameters. Both VS and PD methods were tested driving slow (15-20km/h) and faster (25-35km/h) with the car and finally the PD method was selected. The final best parameters were $k_V = 1$, $k_D = 2$, $ssc_y = 0.5$, $g_P = 1.25$, $g_D = 0.625$, $D_{lim} = 8$. Note that the maximum speed of DJI M600 is 65km/h in calm air that is why EGV speed was limited to 35km/h. Fig. 3 shows real flight the tracking with the final PD method.

The evaluation of the tracking is done considering the tracking errors and the camera coverage of the intersection. The absolute along (e_A) and cross (e_C) errors are calculated only outside the intersection as tracking precision is more important there. The onboard camera is gimbal stabilised so can be considered as downward looking. From the fixed 40m (above ground level) flight altitude the

covered area results as $71 \times 53m$. Considering the coverage of the intersection the time until the whole intersection is in camera field of view (intersection coverage time) should be determined. This is visualized with the dashed and dotted rectangles in Fig. 1. In the evaluation coverage was considered satisfactory when more than 90% of the intersection area was covered by the camera.

Fig. 3. EGV (blue) and forerunner drone (orange) tracks in Smart City. Orange rectangle shows the critical intersection.

The results are summarized in Table 2 with mean absolute along and cross errors and the intersection coverage times (t_1 / t_2). Note that coverage of the critical intersection surrounded by buildings (shown in Fig. 3) was considered driving through it twice. The table shows that the PD method gave smaller mean tracking errors and more consistent coverage times.

Table 2. Real flight test results

Method	e_A [m]	e_C [m]	t_1 [s]	t_2 [s]
VS	4.7	2.8	7.9	4.6
PD	2.3	0.8	6.9	6.4

After in-flight evaluation of the controllers a more rigorous approach should be applied looking for improvement possibilities. This means introduction of possible new methods, evaluation of system stability and the construction of tuning guidelines when possible.

3. FREEDRIVE CONCEPT RE-EVALUATED

Real flights have shown that the buildings can decrease the intersection coverage time because they occlude the view of the camera. So intersection coverage should be increased as much as possible having the drone above the intersection as fast as possible. That's why it is worth to re-evaluate the freedrive concept (introduced in Nagy (2021)). Its equations with VS and PD control are summarized in Table 1. The table shows that inside the intersection the freedrive method is the same as the routedrive, while outside the position error is relative to the IN point and the car velocity is not considered in the control laws as the goal is to fly to the IN point as fast as possible.

4. STABILITY OF THE METHODS

For evaluating the stability of the controllers a dynamical model of the system is required. Thus system identification flight tests of the DJI M600 were conducted applying half doublet tangential (x/T) and normal (y/N) velocity references (3 m/s and 5m/s to avoid pitch/roll angle saturations). Evaluating the results has shown that the tangential (v_T) and normal (v_N) velocity dynamics of the M600 are very similar so one dynamical model is satisfactory. A transfer function (TF) model was determined based-on the flight test data giving (2).

$$G_V(s) = \frac{4.086}{s^2 + 3.3683s + 4.086} = \frac{V_{dj}(s)}{V_{jref}(s)} \tag{2}$$

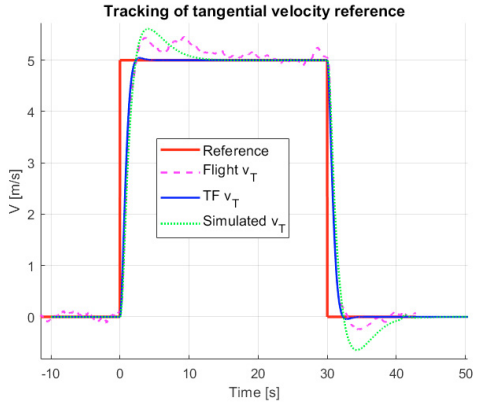

Fig. 4. Tangential velocity tracking comparison

For detailed simulation of the DJI M600 a high fidelity 6DOF model was identified based-on the real flight tests (as mentioned in Section 2). Fig. 4 compares tangential velocity tracking in flight, by the TF model and by the high fidelity simulation model. It shows that while the TF underestimates, the high fidelity model a bit overestimates the overshoots but otherwise the dynamics are very similar. So both TF and high fidelity simulation models can be applied for controller tuning and evaluation.

All control methods give V_{jref} from which V_{dj} can be determined by the transfer function (2) and its integral gives D_{dj}. This is considered in the analysis of all controllers.

4.1 Stability of the velocity scaling control (VS)

As the main dynamics of the EGV tracking is along the road and the tangential and normal M600 dynamics are the same from now on $j = x$ is assumed and j removed. All VS controllers operate with the $k_T(D)Vmax$ distance error dependent control law either adding V_c car velocity or not. So for stability evaluation the dynamics of the position error D should be derived resulting in (3) where a_c is car acceleration, $V_{cx} \approx \|V_c\|$ is assumed and $D_d \geq 0$. $k_T(D)Vmax$ is nonlinear, but can be approximated by a linear expression $k(D) \cdot D$ to determine the maximum tolerable gain of the system.

$$\dot{D} = V_c - V_d - M_O\dot{O} - \dot{M}_O O = \begin{cases} V_c - V_d - \frac{(D_{lim} - \int V_d dt)0.2V_c a_c}{D_{lim}} + \\ \frac{4+0.1V_c^2}{D_{lim}} V_d \\ V_c - V_d + 0.2V_c a_c \end{cases} \begin{matrix} \\ \\ \end{matrix} \begin{matrix} \text{if } D_d \le D_{lim} \\ \\ \text{if } D_d > D_{lim} \end{matrix} \quad (3)$$

With this expression the feedback interconnection of the VS method results as shown in Fig. 5. Note that $1 - 0.2a_c$ is from $\left(1 - \frac{D_{lim}0.2a_c}{D_{lim}}\right) V_c$.

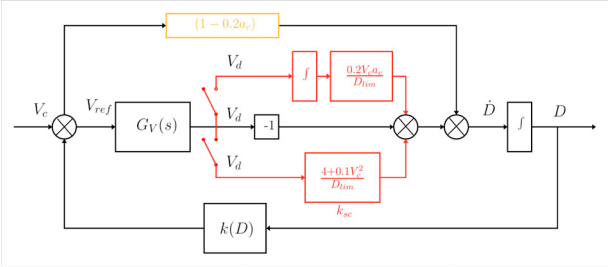

Fig. 5. Interconnection structure of the velocity scaling (VS) control

Based-on the figure the control loops for the different methods and intersection relative positions result as:

- Routedrive outside intersection: full structure
- Freedrive outside intersection: only the black part with $V_c = 0$
- Routedrive inside intersection: black and orange parts without the $0.2a_c$ gain in the orange
- Freedrive inside intersection: the same as for routedrive

The red parts in the scheme are only active when $D_d \in [0, D_{lim}]$. The upper part would give an infinite term in steady state but in this range it is simply a disturbance between 0 and $0.2V_c a_c$ (note the integral term detailed in (3)). The lower red part means an additional gain $k_{sc} = \frac{4+0.1V_c^2}{D_{lim}}$ in the loop so its possible values should be considered. For the low speed tests $V_c \le 10m/s$ meaning maximum $k_{sc} = 13.5/D_{lim}$ gain. Setting $D_{lim} \ge 4.5$ results in $k_{sc} \in [0, 3]$. For fixed values of k_{sc} a root locus for $k(D)$ can be executed. The results (maximum stable $k(D)$ gain) are shown in Table 3.

Table 3. Stable gains for the VS method

k_{sc}	0	0.4	0.8	1	1.5	2	2.5	3
$k(D)$	3.3	2.4	1.85	1.67	1.34	1.12	0.94	0.83

The table shows that the possible feedback gain range decreases as the k_{sc} term increases as expected. Note that the k_{sc} part is only present for a limited time so even if a given k_{sc} is destabilizing the system will converge back to stability. At the next step a relation between $k(D)$ gain and $k_T(D)Vmax$ should be established. The gain is defined pointwise as $k(D) = \frac{k_T(D)Vmax}{D}$, $D \neq 0$ meaning that if the maximum of the right hand expression is below $k(D)$ the system remains stable. Taking the derivative with respect to D gives:

$$\frac{\partial}{\partial D}\frac{k_T(D)Vmax}{D} = -\frac{sign(D)}{2D^2}\left(1 - cos\left(\frac{D\pi}{D_{LIM}}\right)\right)Vmax + \frac{sign(D)}{2D} sin\left(\frac{D\pi}{D_{LIM}}\right)\frac{\pi}{D_{LIM}}Vmax = 0$$

The above equation was numerically evaluated and has a maximum at $D = 0.742D_{LIM}$ and so $max(k(D)) = max\left(\frac{k_T(D)Vmax}{D}\right) = \frac{1.1382Vmax}{D_{LIM}}$. Finally, the selection of D_{lim}, $Vmax$ and D_{LIM} should be done based-on the limitations.

Considering the real flight test at ZalaZONE the maximum EGV velocity was $8.45m/s$ with $D_{lim} = 8$ so $k_{sc} = 1.35$. For this $k(D) \le 1.41$ is required (see Table 3). Considering $Vmax = ssc_x k_V \|V_c\|$ with $ssc_x = 1$, $D_{LIM} = k_D\|V_c\|$ and the $k_V = 1, k_D = 2$ gains leads to $max(k(D)) = \frac{1.1382Vmax}{D_{LIM}} = \frac{1.1382k_V}{k_D} = 0.57$ well below the limit. So the original trial and error tuning resulted in a stable controller.

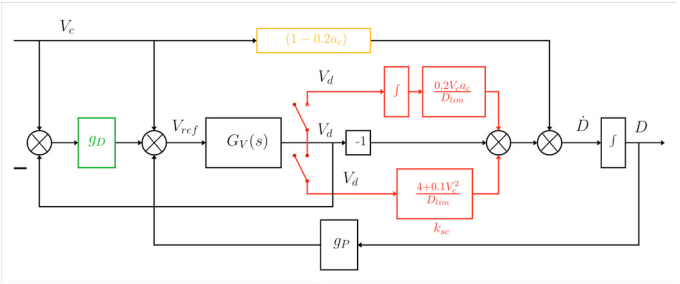

Fig. 6. Interconnection structure of the PD control

4.2 Stability of the PD control

In case of the PD control (again in the x direction) the velocity reference is either $V_{ref} = V_c + g_P D + g_D \Delta V$ or $V_{ref} = g_P D$. The tracking errors are the same as for the VS control but the velocity error $\Delta V_j = V_{cj} - V_{dj}$ is also applied. Considering the same derivations for system dynamics as for the VS control and the PD control law finally, the feedback structure results as shown in Fig. 6.

In the figure again the red parts are temporary when $D_d \in [0, D_{lim}]$, the upper red part is simply a disturbance between 0 and $0.2V_c a_c$ and the lower red part is a gain $k_{sc} \in [0, 3]$ (setting $D_{lim} \ge 4.5$). Again the summary of the four different modes:

- Routedrive outside intersection: full structure
- Freedrive outside intersection: only the black part with $V_c = 0$
- Routedrive inside intersection: black, orange and green parts without the $0.2a_c$ gain in the orange
- Freedrive inside intersection: the same as for routedrive

The figure also shows that this is a cascade connection with tunable gains g_P, g_D and an uncertain gain k_{sc}. Tuning depends also on the working mode of the control.

For freedrive outside the intersection $k_{sc} = 0, g_D = 0$ and so the only tunable parameter is g_P. A root locus shows

that $g_P < 3.36$ gives a stable system and $g_P < 0.85$ gives overshoots below 20%.

For routedrive outside the intersection after fixing g_D a gain sweep should be done for k_{sc} finding the maximum stable g_P gains (and the maximum gains for 20% overshoot) in every case. The results are summarized in Table 4.

Table 4. Stable gains for the PD method in routedrive concept outside the intersection

g_D	k_{sc}	0	0.4	1	1.5	2	2.5	3
0.5	g_P	5	3.6	2.5	2	1.7	1.43	1.25
0.5	g_P 20%	1.4	1	0.7	0.55	0.5	0.4	0.3
1	g_P	6.7	4.8	3.4	2.7	2.2	1.9	1.7
1	g_P 20%	1.7	1.2	0.8	0.7	0.6	0.5	0.4

For both freedrive and routedrive inside the intersection $k_{sc} = 0$ and so tuning of g_P, g_D is required. In the inner loop g_D can be as large as $g_D = 10$ and the loop remains stable. So root locus results with fixed g_D values are shown in Table 5 considering the fact that $g_D < 1$ gives overshoots lower than 10% in the inner loop. The table shows that increasing the g_D gain (making the inner loop faster) increases the stable range of the outer loop and the outer loop gain related to the 20% overshoot.

Table 5. Stable gains for the PD method inside the intersection

g_D	$g_P <$	$g_P <$ for $<$ 20%
0.1	3.69	0.964
0.5	5	1.39
1	6.7	1.66

Considering the real flight test at ZalaZONE the maximum EGV velocity was $6m/s$ and $D_{lim} = 8$ so $k_{sc} = 0.93$. As $g_D = 0.625$ is between 0.5 and 1 from Table 4 $g_P < 2.5$ surely gives a stable system so $g_P = 1.25$ is stable again proving correct trial and error tuning.

4.3 Stability of the mode switching

Besides the stability of the control laws the stability upon outside to inside intersection (and backward) switching should also be verified. There can be jumps in the error signals and even in the control gains but once entering an intersection is declared there is no change in the laws until leaving it which takes several seconds. The same is true for exiting an intersection. So there is no possibility of chattering in the control laws.

5. TUNING OF CONTROLLERS AND TEST RESULTS

To improve the results and evaluate performance the controllers were tuned considering GPS recorded Smart City car tracks and applying the high-fidelity simulation model of the DJI M600. Slow (15-20km/h, log071) and faster (20-25km/h, log072) car tracks were recorded as mentioned before. The same parameters are determined for comparison as in Table 2 for the real flight but also considering absolute along and cross errors inside the intersection.

The different tracking methods are tuned according to the following concepts:

- Routedrive with velocity scaling control: $ssc_y = 0.5$ fixed, $D_{lim} \in [6, 12]$ selected so $k_{sc} \leq 2.25$ and from Table 3 $k(D) \leq 1$ guarantees stability. The latter means $\frac{1.1382Vmax}{D_{LIM}} \leq 1$ and so $D_{LIM} \geq 1.1382Vmax$. Setting $Vmax = \max([k_V \| V_c \|, 2])$ tuning can be done for the k_V parameter. Finally, all combinations of $k_V = 0.25 : 0.25 : 2.5$ and $D_{lim} = 6 : 2 : 12$ were considered.
- Routedrive with PD control: considering the inner loop $g_D \leq 1$ gives 10% or less overshoot so $g_D = 0.5 : 0.1 : 1$ is the selected tuning range. From Table 4 considering again $D_{lim} \in [6, 12]$ and so $k_{sc} \leq 2.25$ the stable range of g_P is about $[1.5, 1.7]$ or below accepting also overshoots above 20%. But as the outer loop 20% gain for $g_D = 1$ is $g_P = 1.7$ inside the intersection (see Table 5) $g_P = 1.5 : 0.1 : 1.7$ is considered. Finally, all combinations of $D_{lim} = 6 : 2 : 12$, $g_D = 0.5 : 0.1 : 1$ and $g_P = 1.5 : 0.1 : 1.7$ were considered.
- Freedrive with velocity scaling control: outside the intersection $Vmax = 15m/s$ is considered which is the maximum dynamic capability of the M600 (that is why the test EGV velocities are below 10m/s). There is no O braking distance and so $k_{sc} = 0$ and $k(D) \leq 3.3$ is all stable. From $\frac{1.1382Vmax}{D_{LIM}} \leq k(D)$ $D_{LIM} = \frac{17}{k(D)}$ so a test campaign for $k(D) = 0.5 : 0.1 : 3$ can be done. 3.3 would give too small distance while $k(D) < 0.5$ a too large one. Inside the intersection the best gains from the routedrive tuning are applied as the control laws are the same.
- Freedrive with PD control: outside the intersection only g_P is applied and again there is no O braking distance and so $k_{sc} = 0$. $g_P < 0.85$ gives less than 20% overshoot so $g_P = 0.5 : 0.1 : 1$ is considered for tuning. Again inside the intersection the beast gains from the routedrive PD tuning are applied.

The best gains resulted as:

- Routedrive with velocity scaling control: $ssc_y = 0.5$, $k_V = 1.5$ and $D_{lim} = 8$
- Routedrive with PD control: $g_P = 1.7$, $g_D = 0.7$ and $D_{lim} = 10$
- Freedrive with velocity scaling control: outside the intersection $k(D) = 0.8$ inside $ssc_y = 0.5$, $k_V = 1.5$
- Freedrive with PD control: outside the intersection $g_P = 0.6$ inside $g_P = 1.7$, $g_D = 0.7$

The best gain results are summarized in Table 6 with the along and cross errors (outside e_A, e_c and inside Ie_A, Ie_C the intersection) and intersection times. The routedrive and freedrive intersection coverage times are about the same irrespective of the control method. However, the freedrive method is better for the slower track (log071) while a bit worse for the faster one (log072). Considering the tracking errors the routedrive method is much better even inside the intersection where both methods track the moving car. The intersection coverage times improved (compared to Table 2) and the performance of the PD methods is more balanced underlining their selection in the demonstration. They are now simulated for the faster

(log072) EGV data and the resulting trajectories are plotted in Fig. 7.

Table 6. Tuning results

Method	LOG	e_A [m]	e_C [m]	Ie_A [m]	Ie_C [m]	t_1 [s]	t_2 [s]
Routedrive VS	071	2.4	0.9	0.6	1	7.9	7.5
	072	3.9	0.9	2.6	1.6	5	4.7
Routedrive PD	071	1.8	0.4	0.2	0.6	7.9	7.4
	072	3.8	0.6	1.5	1	5.1	5
Freedrive VS	071	10	2.7	1.5	1.5	8.3	8.2
	072	11.5	3.2	2	1.8	4.6	4.1
Freedrive PD	071	9.6	1.9	2	0.6	8.3	8.1
	072	11.3	2.5	2	1.3	4.5	4.4

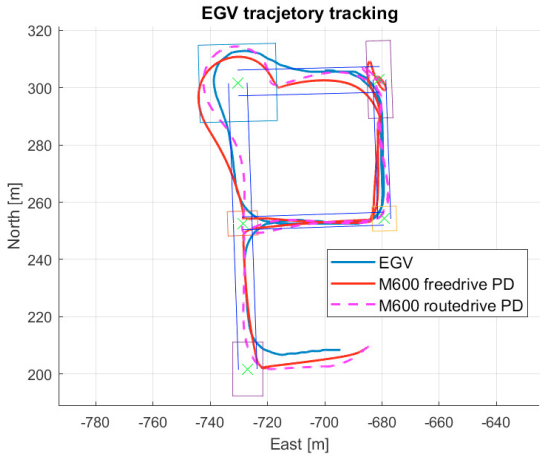

Fig. 7. Tracking of EGV route with different controllers

The figure shows that both tracking methods give smooth results without any sudden maneuvers. It also shows the difference between freedrive and routedrive as in freedrive mode the M600 flies to the next IN point on the shortest route while in routedrive it goes back above the road centerline after the intersection. Possibly this is the cause of the difference between the intersection coverage times as the routedrive method flies a bit longer. If the observation of the EGV surroundings between the intersections is also important than the routedrive PD method is suggested while if only the intersection coverage time is important the freedrive PD method gives better results.

6. CONCLUSION

The paper introduces different tracking methods applicable to track a ground vehicle with a drone flying in front of it but getting closer at intersections to react for sudden route changes. Routedrive and freedrive concepts are introduced the former flying braking distance ahead of the vehicle and going back above it at intersections while the latter flying to the intersection as fast as possible and waiting for the vehicle there. Velocity scaling and PD tracking methods are applied in both concepts. First, in-flight tuning results are demonstrated for the routedrive methods considering demonstrations in the Smart City module of ZalaZONE proving ground (Hungary). Then the stability of the methods is examined and parameter tuning guidelines are provided. Exhaustive tuning of the methods is done based-on real car GPS trajectory data from the demonstrations. Ater the evaluation of the results the suggestion is to use the routedrive PD control method if observation of the road ahead of the EGV and observation of the intersection are equally important. If only the observation of the intersection is important then the freedrive PD control method is suggested.

ACKNOWLEDGEMENTS

This work was supported by the project "Developing innovative automotive testing and analysis competencies in the West Hungary region based on the infrastructure of the Zalaegerszeg Automotive Test Track" GINOP-2.3.4-15-2020-00009.

Part of the research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems. (RRF-2.3.1-21-2022-00002)

Project no. TKP2021-NVA-01 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme.

REFERENCES

Ariyur, K.B. and Fregene, K.O. (2008). Autonomous tracking of a ground vehicle by a UAV. In *2008 American Control Conference*. IEEE. doi: 10.1109/acc.2008.4586569.

Greatwood, C., Bose, L., Richardson, T., Mayol-Cuevas, W., Chen, J., Carey, S.J., and Dudek, P. (2017). Tracking control of a UAV with a parallel visual processor. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. doi: 10.1109/iros.2017.8206286.

Hentati, A.I. and Fourati, L.C. (2020). Mobile target tracking mechanisms using unmanned aerial vehicle: Investigations and future directions. *IEEE Systems Journal*, 14(2), 2969–2979. doi:10.1109/jsyst.2019.2941452.

Hiba, A., Bauer, P., Nagy, M., Simonyi, E., Kisari, A., Kuna, G.I., and Drotar, I. (2022). Software-in-the-loop simulation of the forerunner UAV system. *IFAC-PapersOnLine*, 55(14), 139–144. doi: 10.1016/j.ifacol.2022.07.596.

Nagy, M. (2021). *Development and simulation testing of a forerunner UAV system*. Master's thesis, Budapest University of Technology and Economics.

Vehicle Industry Research Center (2022). SZE drone research: Final forerunner drone demonstration on IV. ZalaZONE Innovation Day. Online article. URL `https://jkk-web.sze.hu/szes-drone-research--final-forerunner-drone-demonstration-on-iv--zalazone-innovation-day/?lang=en`.

Xiao, W., Fan, Q., Li, X., Xiong, Z., and Zhang, J. (2018). Control strategy of ground target tracking for fixed-wing UAV. In *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*. IEEE. doi:10.1109/gncc42960.2018.9018698.