Autonomous Robots manuscript No.
(will be inserted by the editor)

Design and Implementation of a New Teleoperation Control Mode for Differential Drive UGVs

Petter Ögren · Peter Svenmarck · Patrik Lif · Martin Norberg · Nils Emil Söderbäck

Received: / Accepted for publication: 13 Nov. 2013

Abstract In this paper, we propose and implement a new control mode for teleoperated Unmanned Ground Vehicles (UGVs), that exploits the similarities between computer games and teleoperation robotics.

Today, all teleoperated differential drive UGVs use a control mode called *Tank Control*, in which the UGV chassis and the pan tilt camera are controlled separately. This control mode was also the dominating choice when the computer game genre *First Person Shooter* (FPS) first appeared. However, the hugely successful FPS genre, including titles such as *Doom, Half Life* and *Call of Duty*, now uses a much more intuitive control mode, *Free Look Control* (FLC), in which rotation and translation of the character are decoupled, and controlled separately.

The main contribution of this paper is that we replace Tank Control with FLC in a real UGV. Using feedback linearization, the orientation of the UGV chassis is abstracted away, and the orientation and translation of the camera are decoupled, enabling the operator to use FLC when controlling the UGV. This decoupling is then experimentally verified.

The developments in the gaming community indicates that FLC is more intuitive than Tank Control and reduces the well known situational awareness problem. It furthermore reduces the need for operator training, since literary millions of future operators have already spent hundreds of hours using the interface.

P. Ögren, M. Norberg, and N.E. Söderbäck
Department of Aeronautical and Systems Technology, Swedish Defence Research Agency (FOI), Stockholm, Sweden, SE-164 94. E-mail: petter.ogren@foi.se

P. Svenmarck and P. Lif
Department of Information Systems, Swedish Defence Research Agency (FOI), Linköping, Sweden, SE-581 11.

1 Introduction

Today, teleoperated UGVs play an increasingly important role in a number of high risk applications, including bomb demolition, reconnaissance, and search and rescue. In fact, robots similar to the one in Figure 1, have already performed over 20.000 Explosive Ordnance Disposal (EOD) missions in Iraq and Afghanistan, [1]. In both EOD and search and rescue tasks, the *time* needed to carry out the mission is often critical. If the bomb to be disposed is on a timer, every second is valuable. In search and rescue tasks, a building on fire might collapse, and in military EOD, operators sometimes "have only 15 minutes to work before they come under enemy fire", [2]. Furthermore, a large portion of this valuable time is spent gaining situational awareness. In this paper, inspired by developments in the gaming community, we propose and implement an approach aimed at reducing this problem, allowing the operators to focus more on performing the task itself.

A number of studies [3–7] have shown that a large portion of mission time is spent improving the situational awareness of the operator, i.e., improving the mental picture the operator has of where the UGV is relative to the surrounding environment in general and the mission objectives in particular. This is true for EOD and search and rescue tasks, [3] as well as in the military context in general, as shown by the following quote "In Military OperationsÉ rapid action is often considered a tactical necessity... The pace of robot missions is, generally, limited by the operators' ability to gain situational awareness and to control the robot, rather than by its top speed" [7]. These problems are emphasized when the robot moves in cluttered areas near collapsed structures or inside buildings. Poor lighting conditions, as well as limitations on field of view

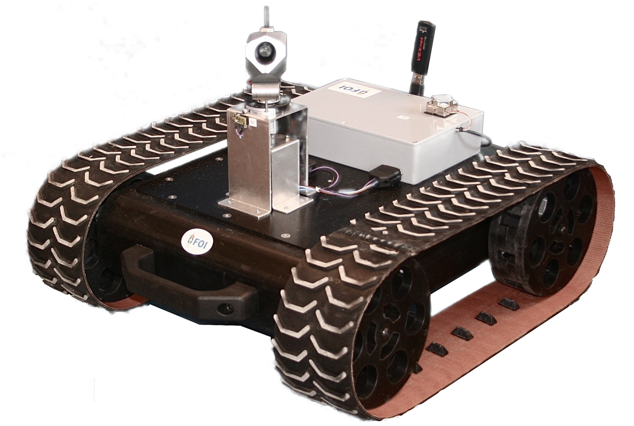

Fig. 1 The UGV in which the proposed control mode was implemented and tested.

and depth perception, makes it harder for the human operator to obtain the necessary situational awareness to search and navigate around obstacles in an efficient manner. In [4], the authors describe how it is often hard for the operator to estimate scales using a video stream, leading to mistakes regarding obstacle sizes and distances. The amount of time spent gaining an appropriate situational awareness is estimated in [5] to as much as 49% and in [6] to roughly 30%.

In this paper, we address the situational awareness problem by improving the way the user interacts with the robot itself, i.e., changing the *control mode* being used. This approach differs from many others, where the sensor fusion part of the problem is investigated, [8]. There are two ways, in which a good control mode can improve the situational awareness of the operator, and hence reduce the mission time. First, it can make it easier to point the camera in the desired direction during robot motion. Second, by reducing the amount of attention needed to control the UGV, it can allow the operator to focus more of the attention on the surroundings of the vehicle and less on the vehicle itself.

The importance of the control mode can also be seen from the developments in the computer game community, where there is a genre called *First Person Shooter* (FPS), including best selling titles such as Quake, Doom, Halo, Half Life, and Call of Duty [9, 10]. A FPS game involves controlling a character through a first person perspective, much like teleoperating a UGV. The character typically moves though a 3D environment, searching for particular items while combating other characters. In the first FPSs, a control mode denoted *Tank Control* was used, which corresponds closely to how todays UGVs are controlled. After a couple of years however, Tank Control was abandoned in favor of so-called *Free Look Control* (FLC)[1]. The basic principle in FLC is to separate translation from rotation. Translation is controlled with one device (joystick 1 or the keyboard) and rotation is controlled with another device (joystick 2 or the mouse). During a transition period, both control modes where included as options in the games, but now FLC completely dominates the market.

In this paper, inspired by developments in the gaming community, we propose to use FLC instead of Tank Control when operating differential drive UGVs with pan-tilt cameras. We also show that this can be done using an extension of the feedback linearization scheme proposed by Lawton *et al.* [11], in a paper where they investigate the problem of autonomously controlling a group of robots moving in close formation. Furthermore, we implement the approach in a real UGV, and experimentally verify the decoupling of camera translation and orientation. This paper extends the theoretical work presented in the conference paper [12].

The outline of this paper is as follows. In Section 2, we formally define Tank Control and FLC. Then, in Section 3, we show how FLC can be applied to a large family of UGV configurations. In Section 4 we describe the advantages and limitation of the approach and in Section 5 we describe how the approach was implemented and verified in a real UGV. Finally, conclusions are drawn in Section 6.

2 Tank Control and FLC

In this section we will give a brief background on FPS games, and then describe and formally define the two control modes: Tank Control and FLC.

FPS games are characterized by an on-screen view that simulates the in-game character's point of view and with a focus on the use of handheld weapons [9], see Figure 2. The first successful FPS games are considered to be Wolfenstein 3D and Doom [9], which appeared in 1993. Both these used Tank Control, which was standard in the genre until 1996 when the game Quake was released. In Quake, there was an option to use another control mode, FLC, and in 1997, with Quake 2, the FLC option was made the default choice [10]. Since then, FLC has totally dominated the genre, with a few notable exceptions that will be discussed below.

We will now give the equations of motion for the two control modes. Let the position of the character be given by $x = (x_1, x_2) \in \mathbb{R}^2$ and the orientation be given by an angle ψ relative to some world fixed coordinate system. Furthermore, let (s_1, s_2) be unit vectors that are rotated an angle ψ, as illustrated in Figure 3. Before

[1] Also known as *Mouse Look Control*.

Fig. 2 Screenshot from the FPS game Half Life 2. Note the weapon at the bottom of the screen, always pointing in the camera direction.

defining the two control modes, we note that the control interface might be either a two joystick gamepad, such as the one in Figure 4, or a keyboard and a mouse. For simplicity, we will assume that it is the former.

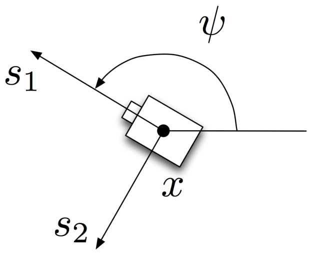

Fig. 3 The character of a FPS game, illustrated by a camera. Note that the coordinate system (s_1, s_2) is fixed to the character.

Fig. 4 A gamepad with two joysticks. Control devices of this type are used for computer games and in recent times also for teleoperation control of robots such as the IRobot Packbot. Furthermore, this gamepad was used to control the UGV in Figure 1.

Below, we will define Tank Control, which dominated the genre for the first couple of years, 1993-1997. The equations of motion are quite straightforward.

Definition 1 (Tank Control)

$$\dot{x}_1 = v_1 \cos\psi, \tag{1}$$

$$\dot{x}_2 = v_1 \sin\psi,$$

$$\dot{\psi} = \omega,$$

where v_1 and ω are given by the inputs from the up-down and left-right motion of the left joystick, c.f. Figure 4.

Note that (1) is also referred to as the Unicycle model [13], and that v_1 corresponds to the velocity in the s_1-direction. Moving on to FLC, we see that it is an extended version of Tank Control, where the possibility to move sideways is added. Writing down the equations of motion we get

Definition 2 (FLC)

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, \tag{2}$$

$$\dot{\psi} = \omega,$$

where v_1 and v_2 are given by the up-down and left-right motion of the left stick, and ω is given by the left-right motion of the right stick.

Note that when $v_2 = 0$, the two models are identical, up to the assignment of control devices. Note further, that the elevation angle of the camera does not influence the character motion and is therefore left out of this paper. Finally, note that v_1 and v_2 are the s_1 and s_2 components of the velocity. In the next section we will see how these models relate to teleoperation of differential drive UGVs.

3 Applying Tank Control and FLC to UGVs

In this section we will first state a kinematic UGV model. We will then show how it can be controlled using both standard Tank Control and the new FLC.

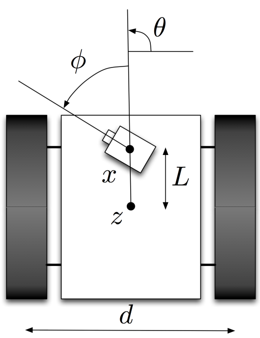

Fig. 5 A general differential drive UGV. The darker rectangles are the tracks, $z = (z_1, z_2)$ denotes the center of the robot and $x = (x_1, x_2)$ denotes the camera position.

Consider the general UGV configuration in Figure 5 and the UGV in Figure 1. A kinematic model of the

UGV can be written as follows.

$$\dot{z}_1 = \frac{v_r + v_l}{2}\cos\theta \tag{3}$$
$$\dot{z}_2 = \frac{v_r + v_l}{2}\sin\theta$$
$$\dot{\theta} = \frac{v_r - v_l}{d}$$
$$\dot{\phi} = k,$$

where $z = (z_1, z_2)$ and θ are the position and orientation of the vehicle, ϕ is the orientation if the camera relative to the vehicle, v_r, v_l are velocities of the right and left tracks respectively, and d is the width of the vehicle. Finally, k is angular velocity of the camera, relative to the vehicle. Note that the camera is mounted a distance L in front of z, giving $x_1 = z_1 + L\cos\theta$ and $x_2 = z_2 + L\sin\theta$. Note that in order to apply Tank Control we will need $L = 0$, whereas we need $L \neq 0$ to apply FLC, as can be seen from the proofs of Lemmas 1 and 2.

If we assume that v_r, v_l, k can be directly controlled by the operator we get the first order model above. Since this model allows instant velocity changes, it is applicable to vehicles with a high force/weight ratio, such as the one in Figure 1.

To see how Tank Control is applied to UGVs we note that the two models are very similar and it is thus quite straightforward to make (3) behave like (1).

Lemma 1 (Tank Control of UGV) *Assuming $L = 0$ and given inputs v_1, ω. If ϕ is set to 0 initially, and the following controls are applied*

$$\begin{pmatrix} v_l \\ v_r \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 \\ -1/d & 1/d \end{pmatrix}^{-1} \begin{pmatrix} v_1 \\ \omega \end{pmatrix} \tag{4}$$
$$k = 0. \tag{5}$$

Then the input to output mapping of model (3) is equivalent to (1).

Proof Since $L = 0$ we have $x = z$. Furthermore, from Figure 5 we have that $\psi = \theta + \phi$. Finally, $k = 0$ gives $\phi \equiv 0$ and $\psi = \theta$ thus, using (6) we get

$$\begin{aligned}\dot{x}_1 &= \dot{z}_1 \\ &= \frac{v_l + v_r}{2}\cos\theta \\ &= v_1\cos\theta \\ \dot{x}_2 &= \dot{z}_2 \\ &= \frac{v_l + v_r}{2}\sin\theta \\ &= v_1\sin\theta \\ \dot{\psi} &= \dot{\theta}, \\ &= \frac{v_r - v_l}{d} \\ &= \omega,\end{aligned}$$

which corresponds to (1). □

Remark 1 Note that $k = 0$ is necessary only when an exact correspondence with (1) is needed. In many UGV applications, Equation (4) is used, but Equation (5) is replaced by letting k be given by the left-right motion of the right stick, giving the advanced user the option of driving in one direction and looking in another.

To apply FLC to a UGV, we need $L \neq 0$, and use an approach that is very similar to the feedback linearization described in [11].

Lemma 2 *If the following controls are applied*

$$\begin{pmatrix} v_l \\ v_r \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 \\ -L/d & L/d \end{pmatrix}^{-1} \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \tag{6}$$
$$k = \omega - \frac{v_r - v_l}{d}.$$

Then the input to output mapping of model (3) is equivalent to (2).

Proof We will now apply feedback linearization, see [14], using the off-axis camera position and orientation, x_1, x_2, as output in a way very similar to the one described in [11]. We start with $x_1 = z_1 + L\cos\theta,\ x_2 = z_2 + L\sin\theta,\ \psi = \theta + \phi$. Differentiating with respect to time and using (3) we get

$$\begin{aligned}\dot{x}_1 &= \dot{z}_1 - L\sin\theta\,\dot{\theta} \\ &= \frac{v_l + v_r}{2}\cos\theta - L\sin\theta\frac{v_r - v_l}{d} \\ \dot{x}_2 &= \dot{z}_2 + L\cos\theta\,\dot{\theta} \\ &= \frac{v_l + v_r}{2}\sin\theta + L\cos\theta\frac{v_r - v_l}{d} \\ \dot{\psi} &= \dot{\theta} + \dot{\phi}, \\ &= \frac{v_r - v_l}{d} + k.\end{aligned}$$

Looking at the camera orientation we note that applying (6) makes

$$\dot{\psi} = \frac{v_r - v_l}{d} + k = \frac{v_r - v_l}{d} + \omega - \frac{v_r - v_l}{d} = \omega$$

as stated in (2). Moving on to $\dot{x}$ and applying the control in (6) we get

$$\begin{aligned}\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \frac{v_l + v_r}{2} \\ L\frac{v_r - v_l}{d} \end{pmatrix} \\ &= R(\theta) \begin{pmatrix} 1/2 & 1/2 \\ -L/d & L/d \end{pmatrix} \begin{pmatrix} v_l \\ v_r \end{pmatrix} \qquad (7) \\ &= R(\theta)R(\phi) \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \\ &= R(\psi) \begin{pmatrix} v_1 \\ v_2 \end{pmatrix},\end{aligned}$$

again as stated in (2). Above we have used the notation $R(\theta)$ to denote a rotation matrix, *i.e.,*

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

This concludes the proof. □

Remark 2 Note that the matrix

$$\begin{pmatrix} 1/2 & 1/2 \\ -L/d & L/d \end{pmatrix}$$

is invertible as long as $L \neq 0$, *i.e.*, as long as the camera is mounted in an off-axis position.

4 Advantages and Limitations of FLC

In this section, we will first see two example executions when using FLC, and then discuss the pros and cons of using the proposed control mode when teleoperating UGVs.

4.1 Two example UGV trajectories when using FLC

Running the algorithm with the kinematic model (3) and the control layer (6), we get the results depicted in Figure 6 and 7.

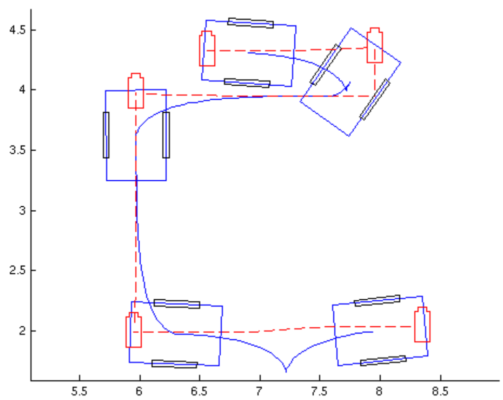

Fig. 6 Resulting camera and vehicle trajectories, when the user commands the translations left, forward, right, forward and left. Note that the camera trajectories are indeed straight line segments.

In Figure 6, the UGV starts out facing eastwards with the camera facing northwards. The user then commands a leftwards (the s_2-direction in Figure 3) motion of the camera by moving the translation stick to the left. Notice how the camera traces out a straight line while the rest of the UGV first backs up, stops, and then moves forward, all the while turning to the left. The user then commands a forwards (the s_1-direction in Figure 3) translation of the camera. Once again, the UGV turns and moves to deliver the desired camera motion. Note also how the camera mounting turns in the opposite direction of the rest of the vehicle to keep the direction of view unchanged. The operator goes on to move the camera rightwards, forwards and finally leftwards again. Again the UGV turns and moves to deliver the commanded straight line motions. Note that these motions are similar to what an operator would do to enhance situational awareness in terms of depth perception, or to get a better view of something that is semi-occluded.

In Figure 7, the UGV is moving towards an object of interest. The operator stops the UGV at roughly a meters distance. He then proceeds to command a translation to the left, while keeping the camera centered on the object. This makes the camera trace out a circular arc at a fixed distance from the object. After having completed a half circle, the operator has seen enough of the object. He then uses the camera orientation joystick to turn the camera in the direction of the first movement, and then goes on to push the translation joystick forwards to make the camera and vehicle continue westwards. These motions could either be used to explore a possible roadside bomb in an EOD mission, or to avoid a hole in the ground in a search and rescue mission.

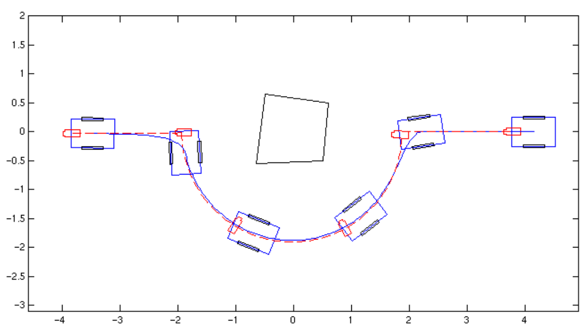

Fig. 7 Vehicle and camera trajectories when passing an obstacle. Note that the commanded left translation results in a circular arc when the camera is kept pointing at the obstacle.

4.2 Developments in the Gaming Community

Three strong reasons for using FLC are given by the developments in the gaming community.

First, as noted in Section 2 above, the computer game industry has gone from only using Tank Control to only using FLC, with a few notable exceptions described below. This change in itself shows that FLC is preferred in an activity that is very similar to UGV teleoperation.

Second, even if the two control modes were equally efficient, it would still make sense to control UGVs in the same way as the majority of the computer games, in order to take advantage of the number of pre-trained operators available. In fact, as noted by Gkikas et al. "There is a large existing expert player community that has developed sensorimotor skills comparable to these of a musical instrument player or an expert typewriter. Actually, one important aspect of game satisfaction for these people is the challenge of achieving mastery in these skills" [9].

Third, by examining the few games that still use Tank Control, we can find additional arguments for switching to FLC. *Resident Evil* is one such game and when asked to explain the reasons why, the producer, Jun Takeuchi, answered as follows: "I think that by imposing certain restrictions on the player you actually help to heighten the fear and the tension, and, ultimately, you create a better horror game." [15]. Thus, in the gaming community, Tank Control is known to heighten the fear and the tension of the user, which makes it highly inappropriate for UGV teleoperation, given the situational awareness problems described in Section 1.

4.3 Hiding chassis orientation

When applying FLC, the attractive feature of decoupling translation and rotation of the camera comes at a price; the chassis orientation θ is hidden from the operator. This fact has two consequences.

First, from a control theoretic perspective, θ, denoted the zero dynamics, must be investigated to make sure that it does not cause any problems in the overall system. In this case, it is straightforward to see that as long as the pan functionality of the camera can rotate freely without any mechanical bounds, using e.g. a slip ring as in the UGV in Figure 1, a growing θ will not cause any problems.

Second, in extremely narrow passages, a given camera motion might cause the UGV chassis to collide with an object. In applications where this is a problem, allowing the operator to switch between Tank Control and FLC might be useful.

4.4 Actuator Saturation and Time Delays

In this section, we will discuss the effects of modeling errors in terms of actuator saturations and time delays on the proposed approach.

The possibility of actuator saturation must always be considered when applying feedback linearization. Obviously, the achievable track velocities are bounded, $v_l, v_r \in (-v_{max}, v_{max})$. Looking at equations (7) we have

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = R(\theta) \begin{pmatrix} 1/2 & 1/2 \\ L/d & -L/d \end{pmatrix} \begin{pmatrix} v_l \\ v_r \end{pmatrix}. \tag{8}$$

Thus, the saturations on v_l, v_r translates to a quadrangle shaped bound centered around the origin on the achievable camera velocities. These bounds make the top speed when using the control layer in Equation (6) dependent on the orientation of the robot chassis. However, this was not perceived to be a major issue when using the UGV in Figure 1, since it was mostly operated at speeds well below its top speed.

Time delays are an important issue in many teleoperation systems, and it has therefore received a lot of attention in the literature. However, in the UGV applications studied in this paper, i.e., EOD and search and rescue tasks, the teleoperation is carried out over short ranges, usually a couple of hundred yards, with very small latencies. Thus, time delays are currently not a major concern for system performance.

If the situation would change, and UGV control over the web became more common, we believe that time delays would actually provide an additional argument for using FLC instead of Tank Control. It is reasonable to assume that orientation errors would cause more severe situational awareness problems than translation errors, and since FLC decouples the two, a rate gyro and a local disturbance rejection control loop can be used to reduce such problems.

5 Implementation and Verification

In this section, we will describe how the proposed approach was implemented in the UGV shown in Figure 1. Furthermore, we will experimentally verify the decoupling of camera translation and orientation that is the essence of FLC.

First, some additions to the scheme presented in Section 3 must be made to account for the modeling errors induced by the slippage of the tracks, [16–18]. These additions are twofold: The kinematic model (3) is slightly extended, and a gyro feeding a Kalman filter is added to measure errors in UGV rotation. An overview of the UGV system can be found in Figure 8.

5.1 Reducing orientation errors

The modeling errors in Equation (3) give rise to errors in both orientation and translation. The former are the

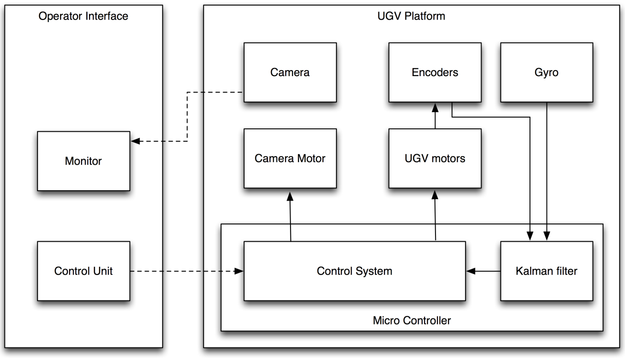

Fig. 8 A System Overview. Note the gyro and motor encoders giving input to a Kalman filter that estimates UGV orientation. The dashed arrows correspond to wireless links.

most serious, as an operator commanding a pure translation towards some object in view of the camera will immediately notice, and be very disturbed by, a large camera orientation error. On the other hand, unless the objects in view are very close, a typical translation error of say 20cm over a few meters of motion will have a quite small impact on the video captured by the camera.

To address orientation errors we equip the UGV with a rate gyro, and use a standard Kalman filter to fuse the signals from that gyro with those of the motor encoders, see Figure 8. As can be seen in Section 5.3 below, this reduces the camera orientation errors to less than 1 % of commanded UGV chassis rotation.

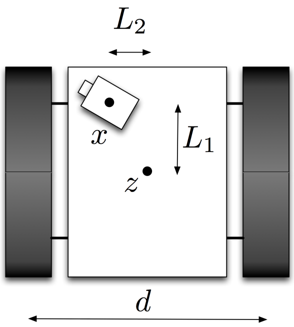

Fig. 9 The new parameters L_1, L_2, d of the UGV model are identified by experiments.

5.2 Reducing translation errors

The translation errors are somewhat harder to reduce, but pose less of a problem, as discussed above. We will apply the approach proposed in [18], which boils down to a slight modification of Equation (3). Instead of measuring the parameters L and d of Figure 5 we now introduce parameters L_1, L_2 and d, as in Figure 9, and identify those parameters from experiments, yielding the results in Table 1

Table 1 Identified UGV parameters.

$d[cm]$	$L_1[cm]$	$L_2[cm]$
70	11	-1.9

Table 2 *Camera drift over five laps, CW rotation of vehicle*

Lap nr	1	2	3	4	5
Accumulated Drift [°]	-3	-4	-4	-5	-6

Table 3 *Camera drift over five laps, CCW rotation of vehicle*

Lap nr	1	2	3	4	5
Accumulated Drift [°]	0	-1	-1	0	+1

However, running the FLC and commanding a straight line camera translation the resulting trajectory overshoots and oscillates heavily when running at high speeds. In order to reduce this phenomenon, d was iteratively reduced until the oscillations ceased at $d = 35$ cm. However, a reduced d makes the UGV turn less sharply, and thus deviates slightly from a commanded straight line translation, as will be shown below.

5.3 Verification of orientation and translation decoupling

To see how well camera orientation was decoupled from translation we commanded the UGV to run multiple laps around a square with a 2m side, each lap resulting in a UGV chassis rotation of 360°. After each lap, the UGV was stopped at its starting position and the camera rotation was measured. The results of running clockwise is found in Table 2 and counter clockwise in Table 3.

As can be seen the drift is about 1 degree per lap when running clockwise and less then 1 degree when running counter clockwise. The asymmetric outcome might be due to asymmetries in hardware or the identified L_2 asymmetry. Either way, a drift of 6 degrees after running 5 laps (1800 degrees) corresponds to 1/3 % which is hardly noticeable by the operator.

To see how well camera translation was decoupled from orientation we used the setup in Figure 10. The UGV chassis was to perform a 90 degree turn to make the camera move along a straight line forwards. After running the UGV for about 2m in the same direction the offset was measured relative to the ideal line of translation running through the original camera position with the same orientation as the camera. The outcomes can be found in Table 4. An average offset of

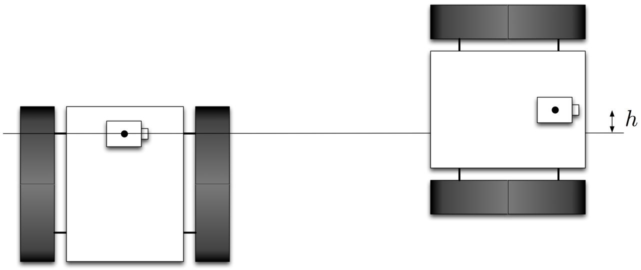

Fig. 10 The setup of the translation decoupling test. The UGV started facing 90 degrees to its right and a pure forward translation was commanded. After about 2m the offset h relative to a straight line in the camera direction was measured.

Table 4 Table of measured offset h

Test no:	1	2	3	4	5	6	7	8	Avg.
h [cm]	-1	0	10	26	32	25	31	20	18

18cm was found. Thus the UGV turns somewhat less sharply than the ideal case, but this can be contributed to the reduction in the d parameter that was needed to reduce oscillations, as described above. To summarize, the rotational decoupling is very accurate, and the translational decoupling is fairly accurate. This shows that it is indeed possible to realize the FLC control mode in a tracked UGV.

6 Conclusions

Teleoperating UGVs is an important, and often difficult and demanding task, which in many ways is similar to playing an FPS computer game. Using a video stream and a gamepad, the operator/user is to control an entity through a 3D scene where situational awareness is often vital for mission success.

Initially, the gaming community used a control mode called Tank Control, that is very similar to how most differential drive UGVs are teleoperated. Currently, however, the gaming community has switched more or less completely to the control mode FLC. This makes it easier for the user to focus on the task at hand by using separate controls to directly position and orient the entity. Inspired by the gaming community, we propose a method where FLC is applied to teleoperated differential drive UGVs. The proposed approach was implemented in a real UGV and the decoupling of camera rotation and translation, that is the essence of FLC, was experimentally verified.

References

1. TALON, "Talon robots – the soldier's choice," *http://www.foster-miller.com/lemming.htm*, 2009.
2. P. A. Buxbaum, "Robot Wars," *Defense Technology International*, March/April 2006.
3. R. Murphy, "Human–Robot Interaction in Rescue Robotics," *IEEE Transactions on systems, Man, and Cybernetics, Part C: Application and Reviews*, vol. 34, no. 2, 2004.
4. D. Woods, J. Tittle, M. Feil, and A. Roesler, "Envisioning Human–Robot Coordination in Future Operations," *IEEE Transactions on systems, Man, and Cybernetics, Part C: Application and Reviews*, vol. 34, no. 2, 2004.
5. J. Burke, R. Murphy, M. Coovert, and D. Riddle, "Moonlight in Miami: An ethnographic study of human-robot interaction in USAR," *Human-Computer Interaction, special issue on Human-Robot Interaction*, vol. 19, pp. 1–2, 2004.
6. H. Yanco and J. Drury, "Where Am I? Acquiring Situation Awareness Using a Remote Robot Platform," *IEEE Conference on Systems, Man and Cybernetics*, 2004.
7. C. Lundberg, "Assessment and evaluation of Man-portable robots for High-risk professions in urban settings," Ph.D. dissertation, Royal Institute of Technology (KTH), Stockholm, Sweden, 2007.
8. A. Saffiotti, D. Driankov, and T. Duckett, "A System for Vision Based Human-Robot Interaction," *IEEE Int. Workshop on Safety, Security, and Rescue Robotics (SSRR-04), Bonn, Germany*, May 2004.
9. K. Gkikas, "The evolution of FPS games controllers: how use progressively shaped their present design," in *Panhellenic Conference on Informatics (PCI)*, 2007.
10. A. Cummings, "The Evolution of Game Controllers and Control Schemes and their Effect on their games," in *The 17th Annual University of Southhampton Multimedia Systems Conference*, 2007.
11. J. Lawton, R. Beard, and B. Young, "A decentralized approach to formation maneuvers," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 6, pp. 933–941, 2003.
12. P. Ögren and P. Svenmarck, "A new control mode for teleoperated differential drive UGVs," in *46th IEEE Conference on Decision and Control*, 2007, pp. 5794–5799.
13. C. de Wit, G. Bastin, and B. Siciliano, *Theory of robot control*. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 1996.
14. S. Sastry, *Nonlinear Systems: Analysis, Stability and Control*. Springer, 1999.
15. M. Grimm, "No change coming to resident evil controls," in *The Escapist Magazine, March*, March 2009.
16. S. A. A. Moosavian and A. Kalantari, "Experimental slip estimation for exact kinematics modeling and control of a tracked mobile robot," in *Proceedings of the 2008 International Conference on Intelligent Robots and Systems*, 2008.
17. D. Endo, Y. Okada, K. Nagatani, and K. Yoshida, "Path following control for tracked vehicles based on slip-compensating odometry," in *Proceedings of the 2007 International Conference on Intelligent Robots and Systems*, 2007.
18. J. Martinez, A. Mandow, J. Morales, A. Garcia-Cerezo, and S. Pedraza, "Kinematic modelling of tracked vehicles by experimental identification," in *Proceedings of the 2004 International Conference on Intelligent Robots and Systems*, 2004.