Cyber-Attacks on Wheeled Mobile Robotic Systems with Visual Servoing Control

Aleksandar Jokic, Amir Khazraei, Milica Petrovic, Zivana Jakovljevic, and Miroslav Pajic

***Abstract*—Visual servoing represents a control strategy capable of driving dynamical systems from the current to the desired pose, when the only available information is the images generated at both poses. In this work, we analyze vulnerability of such systems and introduce two types of attacks to deceive visual servoing controller within a wheeled mobile robotic system. The attack goal is to alter the visual servoing procedure in such a way that mobile robot achieves the pose defined by an attacker instead of the desired one. Specifically, the attacks exploit image transformations developed using a methodology based on simulated annealing. The main difference between the attacks is the considered threat model – i.e., how the attacker has infiltrated the system. The first attack assumes the real-time camera feed has been compromised and thus, the images from the current pose are modified (e.g., during the acquisition or communication); for the second, only the desired destination image is potentially altered. Finally, in 3D simulations and real-world experiments, we show the effectiveness of cyber-attacks.**

I. Introduction

Modern autonomous robotic systems, both in industrial and non-industrial settings, rely heavily on wired or wireless communication for a wide range of applications [1], [2]. Even though this connectivity provides substantial benefits in terms of flexibility and adaptability, it also gives rise to the potential of malicious cyber-attacks [3], [4], [5]. For this reason, the cybersecurity of robotics systems become the focus of significant research efforts [2], [6], [7], [8], [9], [10].

The cyber-attacks that can be utilized to tamper with a robotic system can be classified into the following three categories [11]: (i) denial-of-service (DoS) attacks, (ii) replay attacks, and (iii) deception attacks. DoS attack intercepts communication between sensors/actuators and control system, and disables further data transmission using different mechanisms such as message flooding or resource exhaustion. The replay attack disables current data flow and presents previously collected data patterns to the receiver. Finally, deception attacks include manipulating sensing or actuation data to influence the behavior of the considered system. DoS attacks compromise data availability and can have significant consequences on the system real-time performance but can be relatively easily detected. On the other hand, the deception attacks are considered the most malicious since they can use a plethora of data alterations to directly control robot behavior while remaining stealthy; therefore, this type of attack is in the focus of this paper.

Examples of the developed attacks utilized against robotic applications include attacks on industrial robots with access to robot's configuration files [12], attacks on pick-and-place manipulators [13], spoofing attacks on UAVs (Unmanned Aerial Vehicles) [14], [15], [16], attacks on End-to-End Autonomous Driving Models [17], [18], and DoS attacks on rescue wheeled mobile robots [19] and on surgical robot [20].

Robotic systems rely heavily on visual information for a wide variety of tasks such as control [21], perception [22], decision-making, and planning [23]. A well-established algorithm for direct control of a robotic system by utilizing image information is Visual Servoing (VS) [24], [25], [26]. Robotic systems that most commonly utilize visual servoing as a motion control strategy include industrial manipulators [27], UAVs [28], or service mobile robots [29]. VS of wheeled mobile robots represents a subdomain of VS that requires a specific control law design, which can integrate robot's nonholonomic constraints [30]. Particular constraints for differential drive robots include the restriction in lateral movement of the robot due to the physical nature of the wheels. Consequently, one of the most utilized visual servoing strategies for nonholonomic mobile robots is Position-Based Visual Servoing (PBVS) [31] since it defines errors in 3D space that are directly used for maneuvering. Having that in mind, we use VS strategies, and specifically PBVS as a case study to analyze the potential security threats for this controller type used within the mobile robot domain.

PBVS completely relies on the integrity of the images acquired by a camera during robot motion and image that is presented to a robot as the desired pose. Alteration of these images represents convenient means for different adversaries to change the behavior of the controlled system in a desired, potentially malicious way (e.g., leading to collision with other objects in the environment or delaying correct execution of given tasks). Since a critical starting point in cybersecurity mechanisms development is the analysis of different vulnerabilities and potential attacks on the systems, in this paper we analyze the possibilities for the design of attacks on the current and desired images within PBVS. Specifically, we introduce a methodology to perform geometric transformations of the current and desired images, leading the PBVS controlled wheeled mobile robot to the pose defined by the adversary instead to the desired pose.

This work is sponsored in part by the ONR awards N00014-23-1-2206 and N00014-20-1-2745, AFOSR award FA9550-19-1-0169, as well as by the NSF under CNS-1652544 award and the National AI Institute for Edge Computing Leveraging Next Generation Wireless Networks, Grant CNS-2112562. This material is also based on research supported by mobility Erasmus+ Programme (Key Action 1 - KA107, App ID 9960), Inter-institutional agreement 2019/20 – 2021/22 between the University of Belgrade and the Duke University, and Grant under Contract No. 451-03-47/2023-01/200105 financed by the Ministry of Science, Technological Development and Innovation of the Serbian Government.

Aleksandar Jokic, Milica Petrovic, and Zivana Jakovljevic are with Department of Production Engineering, University of Belgrade – Faculty of Mechanical Engineering, Belgrade 11120, Serbia (e-mail: ajokic@mas.bg.ac.rs, mmpetrovic@mas.bg.ac.rs, zjakovljevic@mas.bg.ac.rs).

Amir Khazraei and Miroslav Pajic are with Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA, (e-mail: amir.khazraei@duke.edu, miroslav.pajic@duke.edu).

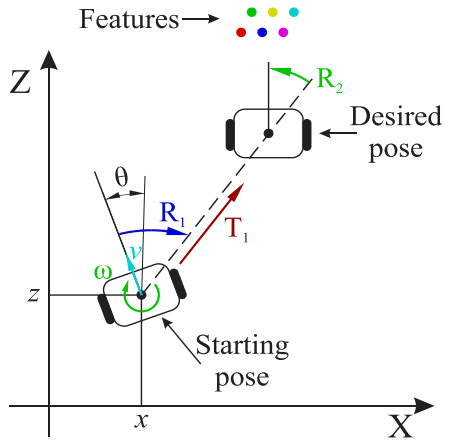

Fig. 1. Mobile robot poses in the external coordinate system.

PBVS requires a methodology for estimation of the difference between the current and desired mobile robot pose in 3D. The homography estimation is one of the most common ways to estimate the error between two poses based on image information. Numerous homography-based PBVS controllers utilized for wheeled mobile robots have been developed in the last decades [27], [28], [34], [35], [36]. Some controllers directly use homography values for control [33], whereas a much more preferred approach is decomposing homography into 3D pose displacement and utilizing that difference for control. Therefore, due to overwhelming efforts devoted to the research within homography decomposition-based PBVS controllers, in this work we focus on vulnerability analysis of such systems.

In particular, to demonstrate their vulnerability, the main contributions of this paper are the development of the deception cyber-attacks on the visual servoing controller implemented within nonholonomic wheeled mobile robotic systems. We present two types of cyber-attacks used to directly alter the desired position that a mobile robot will achieve during visual servoing, which is accomplished by manipulating the images in the current or desired pose. Specifically, we utilize a simulated annealing metaheuristic optimization algorithm to transform the image data and establish pose error (defined and controlled by the attacker) between the desired and achieved positions after the visual servoing is completed; effectively moving the robot to a wrong position fully specified by the attacker.

II. Visual Servoing Controller

In this paper, we consider a wheeled mobile robot with a differential drive system and the unicycle kinematical motion model. The mobile robot pose is defined with the vector $\mathbf{x} = (z, x, \theta)^T$, where z and x are the mobile robot coordinates in the horizontal plane (Fig. 1), and θ represents the steering angle. Moreover, the forward-facing camera is attached to the mobile robot so that the camera coordinate system is the same as the mobile robot's frame, with the offset in the Y direction. The mobile robot moves in the horizontal plane by applying the angular velocities to the wheels to achieve translational (v) and angular (ω) velocity:

$$\begin{bmatrix} \dot{z} \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{1}$$

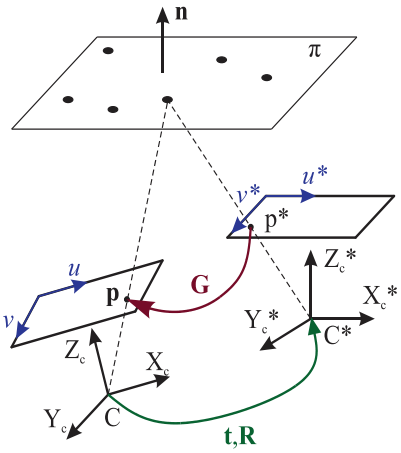

Fig. 2. Homography between the current and desired pose. Camera coordinate systems at the current and desired pose are defined with C and C*, respectively.

In the wheeled mobile robotic domain, visual servoing outputs are velocities required to achieve the desired pose. The desired pose is unknown; the only available information is the desired image.

A. Homography-based visual servoing

Let us consider a set of points belonging to the same plane in the robot environment - π (as illustrated in Fig. 2). Each point in the current image is defined with a vector of homogeneous coordinates $\mathbf{p} = (u, v, 1)^T$ in the image plane, and its corresponding point in the desired image with $\mathbf{p}^* = (u^*, v^*, 1)^T$. Vector of homogeneous image plane coordinate ($\mathbf{p}$) defined in pixels can be transformed to projective normalized coordinates $\mathbf{m} = (x_i, y_i, 1)$ defined in the camera coordinate system by utilizing calibration matrix $\mathbf{K}$:

$$\mathbf{m} = \mathbf{K}^{-1}\,\mathbf{p}, \tag{2}$$

where camera calibration matrix $\mathbf{K}$ is defined by:

$$\mathbf{K} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}; \tag{3}$$

here, α_u and α_v are the focal lengths in pixels, and u_0 and v_0 represent coordinates of the principal point. If the set of minimally four noncolinear points can be detected in both the current and desired image, it is possible to estimate the projective homography matrix $\mathbf{G} \in \mathbb{R}^{3\times3}$ [37] as:

$$\mathbf{p}^* = \mathbf{G}\,\mathbf{p}. \tag{4}$$

Moreover, the projective homography matrix can also be calculated by using relative translation and rotation (Fig. 2) between the camera coordinate frames where the current and desired images are generated:

$$\mathbf{G} = \gamma\mathbf{K}(\mathbf{R}+\mathbf{t}\mathbf{n}^T)\mathbf{K}^{-1}; \tag{5}$$

here, $\mathbf{R} \in \mathbb{R}^{3\times3}$ is the rotation matrix between the camera frames, $\mathbf{t} = (t_x, t_y, t_z)^T$ is a translation vector between the camera frames, $\mathbf{n} = (n_x, n_y, n_z)^T$ is a normal unit vector of the plane π, all three are expressed in the current camera coordinate system (C – Fig 2), and γ is a scale factor.

The estimated projective homography matrix ($\mathbf{G}$) and the calibration matrix ($\mathbf{K}$) can be used to reconstruct the difference (up to a scale) between the poses where the

current and desired images are generated; this can be achieved by utilizing the homography decomposition procedure from [32]. Since the considered mobile robot moves in the plane, it is essential to emphasize that even though homography produces a translation vector with three components and a rotation matrix that is used to compute three Euler angles, only two translation components (in Z and X directions), and rotation angle around Y axis are significantly different from zero. The rest of the parameters exist only due to noise in the feature detection process. The outputs of the decomposition process [32] are four solutions for **n**, **t**, and **R**:

$$\begin{aligned} Rtn_a &= \{\mathbf{R}_a, \mathbf{t}_a, \mathbf{n}_a\}, \\ Rtn_b &= \{\mathbf{R}_b, \mathbf{t}_b, \mathbf{n}_b\}, \\ Rtn_{a-} &= \{\mathbf{R}_a, -\mathbf{t}_a, -\mathbf{n}_a\}, \\ Rtn_{b-} &= \{\mathbf{R}_b, -\mathbf{t}_b, -\mathbf{n}_b\}. \end{aligned} \tag{6}$$

It is important to note that there are two different solutions (*a* and *b*) and their opposite forms. By utilizing the visibility constraint [32], two impossible solutions can be discarded by using (7) for all feature points in the desired image (**m***):

$$\mathbf{m}^{*T}\mathbf{n} < 0. \tag{7}$$

Further, since the mobile robot moves in the horizontal plane, the constraint regarding the constant value of the Y coordinate (dt_{y1} and dt_{y2} represent the difference between initial and current Y coordinate for two possible solutions) can be used to find the index of the true solution:

$$idx = \min([dt_{y1}, dt_{y2}]). \tag{8}$$

The acquired index is utilized to determine the true solutions for **R**, **t**, and **n**. The proposed controller is designed according to the nonholonomic wheeled mobile robot controller that utilizes three steps [33]. The three steps are: (I) rotation to the desired pose (R_1, see Fig. 1), (II) translation to the desired position (T_1, see Fig. 1), and (III) rotation to the desired pose (R_2, Fig. 1). The translational and angular velocities for controller steps are:

$$\begin{aligned} &\text{Step 1}: v = 0,\ \omega = k_{\omega 1} e_1, \\ &\text{Step 2}: v = -k_v e_2,\ \omega = 0, \\ &\text{Step 3}: v = 0,\ \omega = k_{\omega 3} e_3; \end{aligned} \tag{9}$$

where $k_{\omega 1}$, k_v, and $k_{\omega 3}$ are control gains. The first step includes the minimization of the angle the mobile robot needs to rotate to achieve the desired position (see Fig. 3), which can be computed as:

$$e_1 = \theta_d, \tag{10}$$

where $\theta_d = \Delta\theta - \text{atan2}(t_x, t_z)$, $\Delta\theta = -\text{atan2}(-R_{31}, \text{sqrt}(R_{11}^2 + R_{21}^2))$, R_{ij} are elements of **R**, $\Delta\theta$ is the difference in orientation between mobile robot at the current and desired pose. Even though the translation vector is determined up to a scale with the factor γ, the ratio of the relative distance between the current and desired pose will still be the same. The second step includes the translation to the desired position; the control input utilized in this step is the projection of relative distance in the Z direction (t_z) onto the Z axis of the normal unit vector of the plane π (n_z) – i.e.,

$$e_2 = d_d, \tag{11}$$

where $d_d = t_z n_z$.

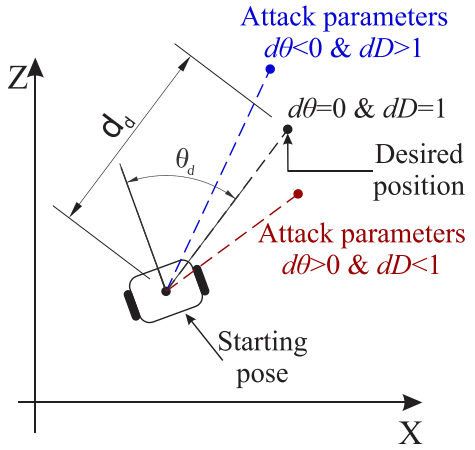

Fig. 3. The influence of the cyber-attack parameters on the mobile robot pose after the visual servoing is completed.

The Z axis of the plane π is utilized since a majority of planes in the human-built environment are vertical and have $n_z \neq 0$. Finally, the third step includes the rotation until the desired pose is achieved. The error is defined by the relative rotation between the current and the desired pose, which can be determined by utilizing rotation matrix **R** since the rotation between coordinate frames is defined as rotation around the Y axis:

$$e_3 = -\Delta\theta. \tag{12}$$

Finally, the derived mobile robot velocity and angular velocity can be utilized to determine the wheel velocities:

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = \begin{bmatrix} 1/r & b/2r \\ 1/r & -b/2r \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}; \tag{13}$$

where ω_r and ω_l represent right and left wheel angular velocity, whereas r is wheel radius, and b is the distance between wheels.

III. Attacks Model and Design

To perform the vulnerability analysis of the PBVS system, our goal is to evaluate whether it is possible to perform specific false-data injection attacks on the images used for control (via a geometric image transformation of the valid images), in order to force the visual servoing controller to converge to the position set by the attacker (i.e., this is the attack objective). The attack parameters, defined by the attacker, include the angle ($d\theta$) and the distance (dD) by which the mobile robot should miss its desired position (as illustrated in Fig. 3). New positions, defined according to the attack parameters (marked with the blue and red circle in Fig. 3) are calculated relative to the desired position, as $d_{d\_new} = dD \cdot d_d$ and $\theta_{d\_new} = d\theta + \theta_d$, capturing the attacker goal to force the mobile robot to, e.g., collide with other entities in the environment or cause a similar accident.

We identify two threat models for the considered mobile robot visual servoing controller (Fig. 4):

- The first model assumes that the attacker has the capability to modify real-time camera images delivered to the controller; such situations may occur when e.g., the camera driver or communication links between the camera and the controller are compromised. We refer to such attacks as Attack #1.

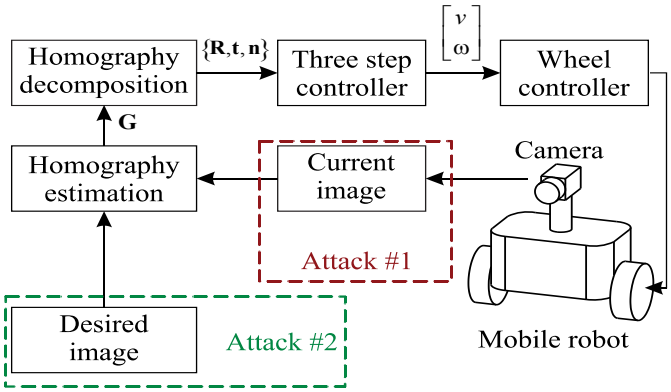

Fig. 4. Flowchart for two types of cyber-attacks.

- The second model considers a misinformation attack, where the attacker is able to modify the desired image provided to the control system at the beginning of the robot move (we refer to this as Attack #2). Such an attack does not require the continuous injection of false camera images.

To demonstrate the system vulnerability, we design the attacks on visual servoing by transforming the current image **I** or the desired image **I***, in order to manipulate the position of image features. A cyber-attack is considered effective if its transformation of the feature position in the image space affects the homography estimation and, therefore, the entire visual servoing process.

In this paper, we consider two separate geometric transformations defined with two parameters. The first transformation is image translation in the u direction by a certain number of pixels defined with parameter T_x. The second transformation utilizes a scaling parameter s; if $s > 1$ the image is centrally cropped and rescaled back to its original resolution, otherwise if ($s < 1$), a zero padding technique is done and the image is rescaled back to its original resolution. The entire transformation matrix with these two parameters is denoted with **T**. Transformed position of each pixel ($\hat{\mathbf{p}}$) in the image can be modeled as:

$$\hat{\mathbf{p}} = \mathbf{T}\mathbf{p}\,, \tag{14}$$

$$\begin{bmatrix}\hat{u}_j\\ \hat{v}_j\\ 1\end{bmatrix} = \begin{bmatrix}1/s & 0 & \left(T_x+(s-1)u_0\right)/s\\ 0 & 1/s & \left((s-1)v_0\right)/s\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}u_j\\ v_j\\ 1\end{bmatrix},\ j=1,\ldots,H\cdot W\,; \tag{15}$$

where H and W represent the total number of pixels in both directions of the image. The 3D parameters the attacker defines ($d\theta$ and dD) need to be correlated with the parameters T_x and s, which is achieved from the following optimization problem:

$$\text{minimize}\quad f(\mathbf{X})=\text{e}\,, \tag{16}$$

$$e=\left\|\hat{\theta}_d-\left(\theta_d+d\theta\right)\right\|_2+\left\|\hat{d}_d-\left(d_d\cdot dD\right)\right\|_2\cdot w\,, \tag{17}$$

$$\mathbf{X}=\left(T_x,s\right); \tag{18}$$

here, $\hat{\theta}_d$ and $\hat{d}_d$ are extracted from the transformed image according to the procedure that was introduced in Section II. Since the distance (d_d) is in meters and the angle (θ_d) in degrees, an additional weight w ensures that both parameters have an equal stake in the optimization process.

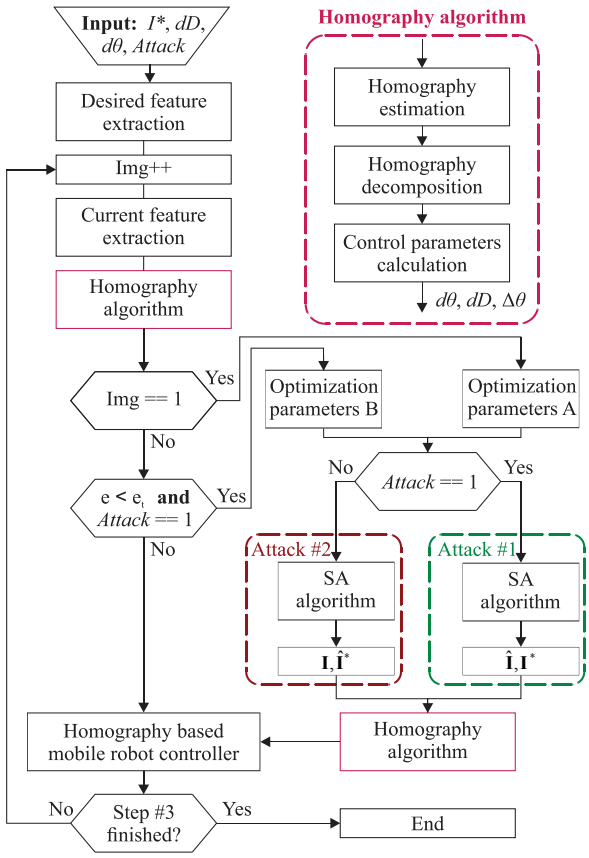

Fig. 5. Detailed flowchart of the cyber-attacks within visual servoing framework.

The detailed algorithm for both attacks is shown in Fig. 5. First, the features are extracted in the provided desired image. We use the center coordinates of the spheres that are visible in the camera's field of view as the features. When the current image is acquired, the features are extracted and matched with the ones in the desired image. The positions of the matched features represent the input into a homography algorithm that performs the homography matrix estimation, decomposition, and control parameters calculations.

Afterwards, the selected cyber-attack is executed if the current image is the first one (Img == 1 in Fig. 5) utilized within the visual servoing algorithm. The main difference between the two proposed attacks is that for Attack #1 there is a continuous real-time modification of the current image acquired during the robot operation. Thus, for this attack an additional optimization procedure is performed if the error e (17) becomes larger than the error threshold e_t. The first optimization process that is performed at the start of both attacks utilizes the set of parameters A (see Fig. 5): bounds are set to ±50 for T_x, ± 0.2 for s, fast annealing function is utilized, and the initial temperature is 100 [38]. An additional optimization for Attack #1 is defined with the set of optimization parameters B, where the bounds of the optimization process are significantly lower and are set to ±2 for T_x, and ± 0.01 for s. Moreover, the optimization process starts from the previously acquired optimal parameters for T_x and s, and the maximum time is set to 0.02s (since this optimization is performed online during visual servoing). Both optimization processes are performed using the Simulated Annealing (SA) algorithm [38].

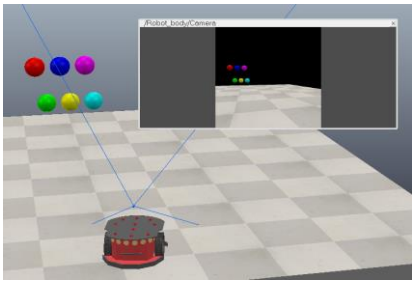

Fig. 6. Mobile robot at the starting pose in simulation environment with camera view window.

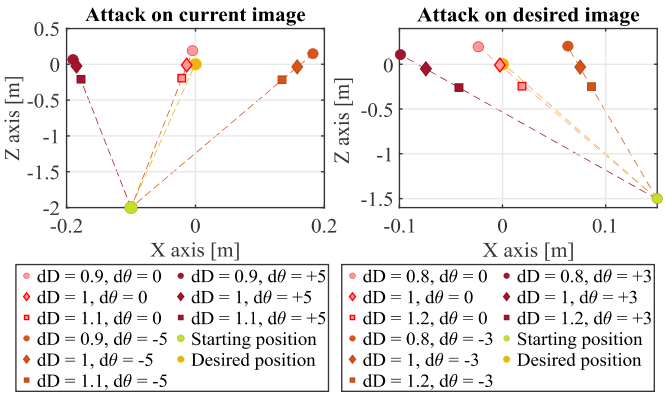

Fig 7. Achieved poses for both simulation experiments for different values of dD and $d\theta$; note that the attack is not present for $dD = 1$ and $d\theta = 0$ – Fig. 4.

IV. Experimental Evaluation

In this section, we illustrate the effectiveness of the proposed attacks on visual servoing controller both in simulation and on a real-world mobile robotic system.

A. Simulation

The 3D simulation is performed within *CoppeliaSim* (V-*REP*) simulator [39] with a standard pioneer p3dx mobile robot equipped with an additional camera sensor. Six detectable coplanar spheres are added in the simulation (Fig. 6) whose positions are unknown to the robot.

Color thresholding is performed to extract the center coordinates of each sphere, used as features in the visual servoing controller. Two experiments for each attack type are executed in simulation with nine different attack parameters; results are summarized in Fig. 7. The mobile robot is set to a target pose $\mathbf{x}_t = (0, 0, 0)^T$, and the desired image is generated and saved. Then, for the attack on the desired image (Attack #2), the mobile robot is set to an initial pose of $\mathbf{x}_d = (–1.5, 0.15, 10)^T$, while the initial pose for an attack on the current image (Attack #1) is $\mathbf{x}_c = (–2, –0.1, –20)^T$; the poses are expressed in meters and degrees. Camera resolution is set to 256 × 256 px, and the overall system has a sampling time of around 0.075s. In the simulation, the camera calibration is assumed to be ideal, with the principle point at the center of the image and no lens distortion. Fig. 6 shows the mobile robot in the initial pose with a camera view and the features.

The positions that the mobile robot achieves with different attacks and parameters are shown in Fig. 7. As can be seen, the attack parameters dD and $d\theta$ significantly alter the final positions of the mobile robot.

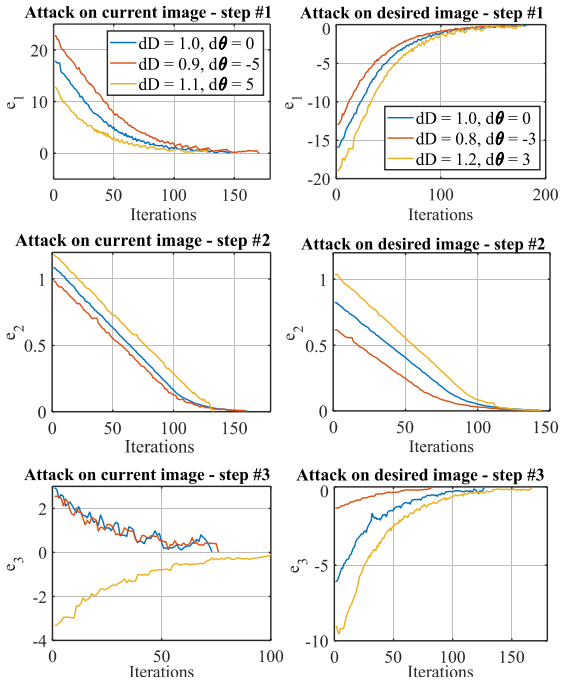

Fig 8. The convergence curves of the errors for all three steps of the visual servoing algorithm. The left and right columns of the figures represent the attacks on current and desired images, respectively.

For the visual servoing without an attack (i.e., $dD = 1$ and $d\theta = 0$), the achieved pose is close to the target one ($\mathbf{x}_t = (0, 0, 0)^T$), whereas attacks with every other considered combination of parameters accurately influence the mobile robot to reach a pose different than the targeted one – e.g., Attack#1 with parameters $dD = 1$ and $d\theta = \pm 5$ results in the robot ending in the poses that vary from $\mathbf{x}_{a1} = (–0.04, 0.16, 0.09)^T$ to $\mathbf{x}_{a2} = (–0.21, –0.18, 0.06)^T$. Thus, the robot can be moved into different poses along the X axis by changing the angle $d\theta$. The same holds for the Z axis (by changing dD) and Attack #2. Note that depending on the attack type (Attack #1 or #2), the sign of the angle parameter changes.

The error values e_1-e_3 from (10)-(12) obtained during both attacks for each of the three steps of mobile robot controller – (9) are shown in Fig. 8. Out of 9 experimental evaluations, the errors for three representative ones are shown. The errors at each step are directly correlated to the values of the attack parameters. Hence, the error values also show the effect of the proposed attacks on the visual servoing controller. Moreover, the error e_3 has an initial negative value within the attack on the current image with $dD = 1.1$ and $d\theta = 5$. This can be attributed to the fact that the mobile robot moves to the left for that experiment, whereas it moves right in the other two experimental evaluations (see Fig. 7).

B. Experimental Evaluation

We demonstrate the effectiveness of the proposed cyber-attacks in real-world settings on two experiments with the mobile robot RAICO (Robot with Artificial Intelligence based COgnition), equipped with Basler Dart daA1600-60uc camera. Images with a resolution of 800 × 600 px are acquired at a rate of 0.075s. The camera calibration matrix is

$$\mathbf{K} = \begin{pmatrix} 433.8 & 0 & 404.8 \\ 0 & 434.9 & 287.9 \\ 0 & 0 & 1 \end{pmatrix}.$$

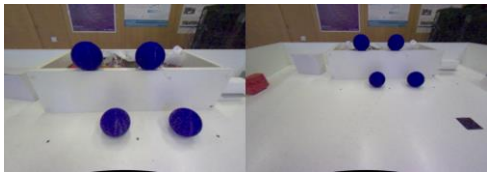

Fig. 9. Target (left) and starting (right) image generated by the RAICO robot.

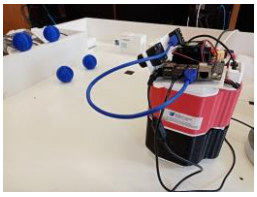

Fig. 10. Mobile robot RAICO at the starting pose during real-world experimental evaluation.

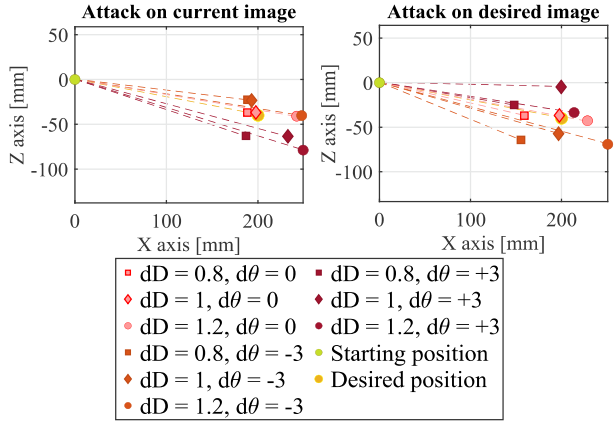

Fig. 11. Achieved positions for both real-world experiments.

The entire visual servoing controller is implemented using the Jetson Nano development kit with Python 3.6.9 and OpenCV 4.1.1. Examples of images generated at the starting and the target pose by the mobile robot are shown in Fig. 9. The initial pose for both experiments was set to be $\mathbf{x}_i = (0, 0, 0)^T$, and the target pose was $\mathbf{x} = (0.20, 0.04, 0)^T$.

Fig. 10 illustrates the experimental setup. The positions the robot achieved with different attack parameters are shown in Fig. 11. According to Fig. 11, the final reached positions of the mobile robot are coherent with the provided attack parameters. As it can be seen, dashed lines in both Fig. 7 and Fig. 11 show mobile robot paths between starting and final poses. The achieved accuracy for the considered experiment without an attack is within ±4mm. The robot paths during two real-world experiments for the two attacks are shown in Fig. 12. Poses are computed according to the wheel encoder data and dead-reckoning pose calculation method [40]. The circle defines the mobile robot's position, while the heading angle is represented with a straight line.

The errors for each step in the controller with different attack parameters are shown in Fig. 13. As shown, the error values under Attack #2 (attack on the desired image) have roughly the same convergence properties. However, for Attack #1 (attack on the current image) with parameters $dD = 1.2$ and $d\theta = -3$ error value converges faster compared to other two experiments. The difference in convergence is due to an additional optimization process that is performed in the case that error e becomes too large (see Fig. 5).

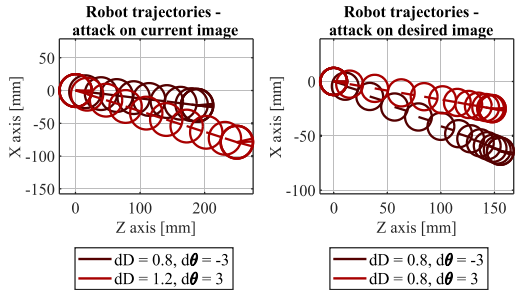

Fig. 12. Achieved positions for both real-world experiments.

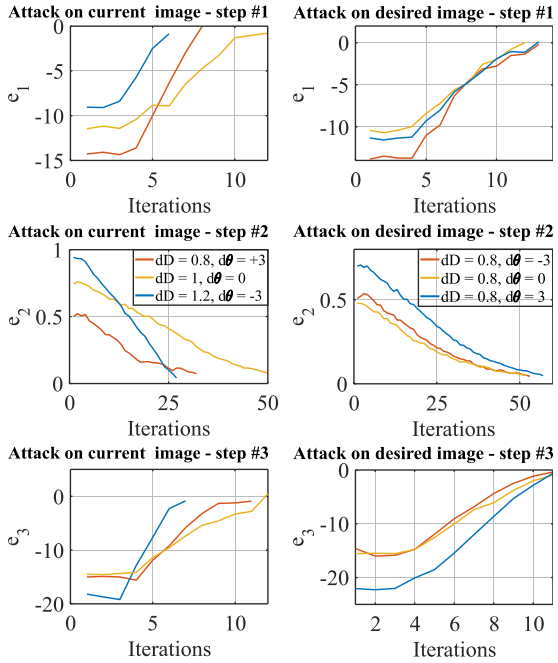

Fig. 13. Achieved errors for both real-world experiments. The left and right columns of the figures represent the attacks on current and desired images, respectively.

Therefore, the optimization process reduces the error e and all three individual errors (e_1, e_2, and e_3). An additional optimization is necessary to ensure that the mobile robot reaches the position defined by the attack parameters.

V. Conclusion

In this work, we have demonstrated the vulnerability of visual servoing control for wheeled mobile robotic systems by introducing a methodology to design effective false-data injection attacks on images used for control. We have considered two threat models where the attacker is able to modify runtime camera images or impact captured image at the desired robot pose. The proposed cyber-attacks utilize an image transformation procedure to alter the final pose, allowing attackers to specify the distance and angle for which the mobile robot will miss its target pose. We have shown that adequate transformation parameters can be acquired via simulated annealing optimization. In simulation and real-world experiments, we have shown the effectiveness of attacks for both threat models, by significantly changing the reached (i.e., final) mobile robot pose both in the X and Z directions. The future research directions include the development of a deep learning-based intrusion detection system for vision-based mobile robot controllers.

REFERENCES

[1] J. E. Rubio, C. Alcaraz, R. Roman, and J. Lopez, "Current cyber-defense trends in industrial control systems," *Computers & Security*, vol. 87, p. 101561, 2019.

[2] A. Khalid, P. Kirisci, Z. H. Khan, Z. Ghrairi, K.-D. Thoben, and J. Pannek, "Security framework for industrial collaborative robotic cyber-physical systems," *Computers in Industry*, vol. 97, pp. 132–145, 2018.

[3] A. Chowdhury, G. Karmakar, and J. Kamruzzaman, "Survey of recent cyber security attacks on robotic systems and their mitigation approaches," in *Detecting and Mitigating Robotic Cyber Security Risks*, IGI global, 2017, pp. 284–299.

[4] T. M. Chen and S. Abu-Nimeh, "Lessons from stuxnet," *Computer*, vol. 44, no. 4, pp. 91–93, 2011.

[5] A. K. Bozkurt, Y. Wang, and M. Pajic, "Secure planning against stealthy attacks via model-free reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 10656–10662.

[6] E. Fosch-Villaronga and T. Mahler, "Cybersecurity, safety and robots: Strengthening the link between cybersecurity and safety in the context of care robots," *Computer Law & Security Review*, vol. 41, p. 105528, 2021.

[7] G. W. Clark, M. V Doran, and T. R. Andel, "Cybersecurity issues in robotics," in *2017 IEEE conference on cognitive and computational aspects of situation management*, 2017, pp. 1–5.

[8] J.-P. A. Yaacoub, H. N. Noura, O. Salman, and A. Chehab, "Robotics cyber security: Vulnerabilities, attacks, countermeasures, and recommendations," *International Journal of Information Security*, pp. 1–44, 2021.

[9] M. Pajic, J. Weimer, N. Bezzo, O. Sokolsky, G. J. Pappas, and I. Lee, "Design and implementation of attack-resilient cyberphysical systems: With a focus on attack-resilient state estimators," *IEEE Control Systems Magazine*, vol. 37, no. 2, pp. 66–81, 2017.

[10] N. Bezzo, J. Weimer, M. Pajic, O. Sokolsky, G. J. Pappas, and I. Lee, "Attack resilient state estimation for autonomous robotic systems," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 3692–3698.

[11] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.

[12] F. Maggi, D. Quarta, M. Pogliani, M. Polino, A. M. Zanchettin, and S. Zanero, "Rogue robots: Testing the limits of an industrial robot's security," *Trend Micro, Politecnico di Milano, Tech. Rep*, pp. 1–21, 2017.

[13] Z. Jakovljevic, V. Lesi, and M. Pajic, "Attacks on distributed sequential control in manufacturing automation," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 775–786, 2020.

[14] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[15] K. Kim, S. Nalluri, A. Kashinath, Y. Wang, S. Mohan, M. Pajic, and B. Li, "Security analysis against spoofing attacks for distributed UAVs," Workshop on Decentralized IoT Systems and Security (DISS), 2020.

[16] A. Khazraei, M. Haocheng, and M. Pajic, "Stealthy perception-based attacks on unmanned aerial vehicles," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[17] A. Boloor, X. He, C. Gill, Y. Vorobeychik, and X. Zhang, "Simple physical adversarial examples against end-to-end autonomous driving models," in *2019 IEEE Int. Conference on Embedded Software and Systems (ICESS)*, 2019, pp. 1–7.

[18] A. Khazraei, H. Pfister, and M. Pajic, "Resiliency of Perception-Based Controllers Against Attacks," in *Learning for Dynamics and Control Conference*, 2022, pp. 713–725.

[19] T. Vuong, A. Filippoupolitis, G. Loukas, and D. Gan, "Physical indicators of cyber attacks against a rescue robot," in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops*, 2014, pp. 338–343.

[20] T. Bonaci, J. Yan, J. Herron, T. Kohno, and H. J. Chizeck, "Experimental analysis of denial-of-service attacks on teleoperated robotic systems," in *Proceedings of the ACM/IEEE 6th Int. Conference on Cyber-Physical Systems*, 2015, pp. 11–20.

[21] A. Paolillo, M. Nava, D. Piga, and A. Giusti, "Visual Servoing with Geometrically Interpretable Neural Perception," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 5300–5306.

[22] V. R. Kumar *et al.*, "Omnidet: Surround view cameras based multi-task visual perception network for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2830–2837, 2021.

[23] A. Jokić, M. Petrović, and Z. Miljković, "Mobile robot decision-making system based on deep machine learning," in *9th International Conference on Electrical, Electronics and Computing Engineering (IcETRAN 2022)*, 2022, pp. 653–656.

[24] J. Huh, J. Hong, S. Garg, H. S. Park, and V. Isler, "Self-supervised Wide Baseline Visual Servoing via 3D Equivariance," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2227–2233.

[25] Q. Bateux, E. Marchand, J. Leitner, and F. Chaumette, "Training Deep Neural Networks for Visual Servoing," in *2018 IEEE Int. Conference on Robotics and Automation (ICRA)*, 2018, pp. 1–8.

[26] A. Jokić, M. Petrović, and Z. Miljković, "Semantic segmentation based stereo visual servoing of nonholonomic mobile robot in intelligent manufacturing environment," *Expert Systems with Applications*, vol. 190, p. 116203, 2022.

[27] Z. Miljković, M. Mitić, M. Lazarević, and B. Babić, "Neural network Reinforcement Learning for visual control of robot manipulators," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1721–1736, 2013.

[28] H. Xie, A. F. Lynch, K. H. Low, and S. Mao, "Adaptive output-feedback image-based visual servoing for quadrotor unmanned aerial vehicles," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 1034–1041, 2019.

[29] M. Petrović, A. Jokić, Z. Kulesza, and Z. Miljković, "Deep learning of mobile service robots," in *Book Service robots – Advances in Research and Applications*, Nova Science Publishers, New York, 2021, pp. 77–97.

[30] X. Liang, H. Wang, and W. Chen, "Adaptive image-based visual servoing of wheeled mobile robots with fixed camera configuration," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6199–6204.

[31] F. Chaumette and S. Hutchinson, "Visual servo control Part 1 : Basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[32] E. Malis and M. Vargas, "Deeper understanding of the homography decomposition for vision-based control," *[Research Report] RR-6303*, INRIA, p. 90, 2007.

[33] G. López-Nicolás, N. R. Gans, S. Bhattacharya, C. Sagüés, J. J. Guerrero, and S. Hutchinson, "Homography-based control scheme for mobile robots with nonholonomic and field-of-view constraints," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 4, pp. 1115–1127, 2010.

[34] S. Benhimane and E. Malis, "Homography-based 2D visual servoing," in *Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2397–2402.

[35] N. Wang and H. He, "Adaptive homography-based visual servo for micro unmanned surface vehicles," *The International Journal of Advanced Manufacturing Technology*, vol. 105, no. 12, pp. 4875–4882, 2019.

[36] N. Wang and H. He, "Dynamics-level finite-time fuzzy monocular visual servo of an unmanned surface vehicle," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 11, pp. 9648–9658, 2019.

[37] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Second Edi. Cambridge University Press, 2000.

[38] Y. Xiang, S. Gubian, B. Suomela, and J. Hoeng, "Generalized simulated annealing for global optimization: the GenSA package.," *The R Journal*, vol. 5, no. 1, p. 13, 2013.

[39] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1321–1326.

[40] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms In MATLAB®*. Springer, 2017.