2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Predictive Tracking Control of a Camera - Head Mounted Display System subject to Communication Constraints

Markus Kögel, Petar Andonov, Marco Filax, Frank Ortmeier and Rolf Findeisen

Abstract— **Remote operation of devices and monitoring can be supported by a combination of an actuated remote camera and a head-mounted display. This work considers the tracking control of a camera - head mounted display system over a communication network subject to delays. The objective is to design a control scheme such that the mismatch between the viewing direction of the head-mounted display and the camera is minimized. For the considered system the camera has a field of view and resolution which is larger than the one of the head-mounted display. The controller at the camera side guarantees that the mismatch of the viewing directions is below a specific bound. Furthermore, the remaining pointing mismatch is compensated by selecting and displaying the part of the image on the head-mounted display corresponding to the desired field of view. The proposed controller employs a prediction model. It is based on set-based methods to take the unknown future movement of the head into account. We outline conditions to guarantee bounded tracking errors in the presence of communication delays. Additionally, we outline how unnecessary communications and actuations can be avoided. The results are illustrated by simulations.**

I. Introduction and motivation

Many control systems and applications operate over communication networks giving rise to so called networked control systems or cyber physical systems [1]–[4]. Utilizing wireless communication networks provides flexibility and decreases hardware cost. However, it also lead to several challenges due, e.g., communication delays or packet dropouts [5], or to new objectives such as the desire to reduce communications [6]. For various applications and network effects there exist solutions by now, see e.g. [1]–[4], [6], [7].

Here we consider the operation and control of a head-mounted display (HMD) and a camera system over a network, see Figure 1. Our objective is that the images displayed match the motion of the head as good as possible, i.e. the field of view matches the one of the human in order to avoid image distortions and discomfort. Preventing a mismatch between motion and vision helps to mitigate certain types of motion sickness [8]. Such systems are used, e.g. for video surveillance, wildlife viewing or within tele-presence system, [9]–[11]. For such applications the communications can be limited or significant delays might be present, especially for unreliable 3G, 4G or other wireless networks.

By now many results for related problems and applications exists. The estimation of head movements has been studied in the context of virtual reality, c.f. [12], [13]. Much work exists for the bilateral tele-operation, which aims at providing a human operating a machine/robot remotely by a manipulator, e.g. a joystick, with haptic feedback, see [14]. Model predictive control (MPC) [15] for constrained systems with delays has been investigated, e.g. in [4], [16]–[20]. Constrained tracking using model predictive control has been studied for example for set-points [15], [21], [22], periodic references [23], references generated by linear models, [24] or well in advance known references [25]–[27].

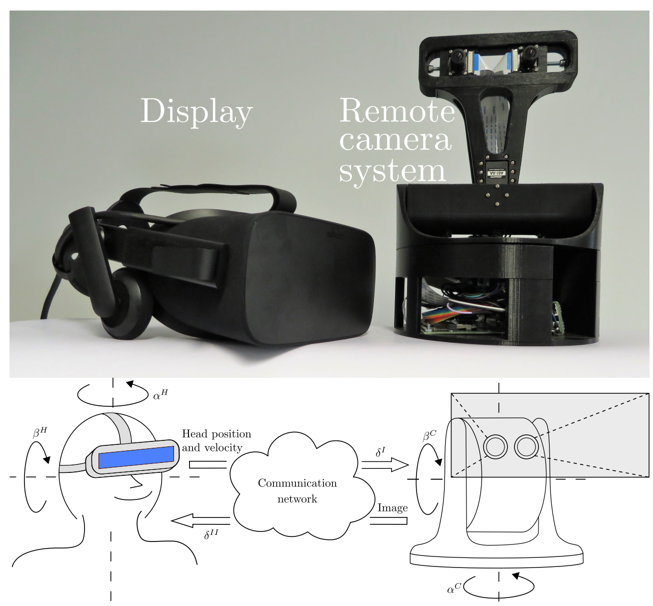

Fig. 1. Proposed control setup

In the considered setup the viewing direction of the human head is measured. Based on the received information it is decided how to move the camera and when images are send to the HMD. The communication to and from the camera is subject to delays. Furthermore dynamics/limitations of the camera actuation such as maximum speed need to be taken into account. Thus a perfect tracking, i.e. achieving that the camera and the human head always points into the same direction is challenging.

The contribution of this work is a control approach to "compensate" the delay and minimize the pointing error and to avoid image distortions. For the approach cameras with

MK, PA, RF are with the Labratory of System Theory and Control at the Otto-von-Guericke-University Magdeburg, Germany and the International Max Planck Research School (IMPRS-ProEng), Magdeburg. MF, FO are with the Chair of Software Engineering, at the Otto-von-Guericke-University Magdeburg, Germany. {markus.koegel, petar.andonov, marco.filax, frank.ortmeier, rolf.findeisen}@ovgu.de.

The authors acknowledge funding by the German research foundation in the frame of the focal research program SPP 1914 and by the EU-programme ERDF (European Regional Development Fund) within the research center of dynamic systems (CDS).

978-3-9524-2699-9 ©2018 EUCA

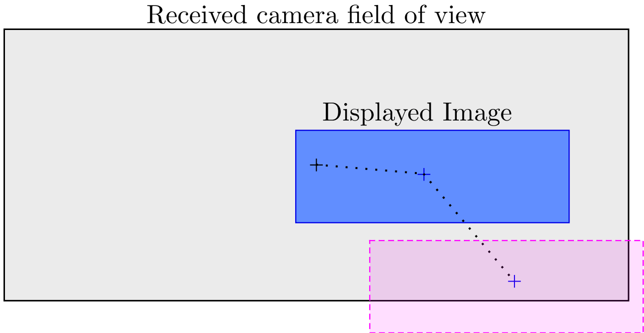

Fig. 2. Illustration for one eye: received/camera image (black), image displayed at HMD (blue) with correct compensation, distorted image (red)

a larger field of view than the image in the HMD are used, see Figure 2.

The main idea is to predict the uncertain future head movement and to choose the camera's input to track the prediction taking uncertainties into account. By achieving a small enough mismatch between the camera and future head orientation it is possible that the HMD can show a corrected image using rather simple image transformations, [28]. For the controller design and analysis we utilize methods from robust/constrained control [15], [29].

Conditions are provided guaranteeing that the viewing direction of the head matches the direction of the displayed image, i.e. a "loss of sight" is avoided. Moreover, extensions are outlined such that images are only transferred if necessary, based on ideas from event-triggered control, [4], [6].

The remainder of this work is structured as follows. Section II presents the detailed problem setup and formulates the considered control problem. In Section III the basic controller design is presented. Section IV outlines extension to reduce the network load and input utilization. Section V outlines simulation results. At the end we provide a summary.

In the following: $\oplus$, $\ominus$ denotes the Minkowski sum/Minkowski difference; a set $\mathbb{E}$ is a robust positive invariant set for the dynamics $e_{k+1} = Fe_{k+1} + d_k$, $d_k \in \mathbb{D}$, where $\mathbb{D}$ is a convex, compact set with $0 \in \mathbb{D}$, if for all $e_k \in \mathbb{E}$, $d_k \in \mathbb{D}$: $e_{k+1} \in \mathbb{E}$, [15], [29]. For a vector x $x(i)$ denotes the ith entry of x.

II. Problem Setup

Next we describe the problem and the utilized models.

A. Real setup - Figure 1

The used HMD is a Oculus Rift with a resolution of 1080×1200 per eye connected to a standard computer. The camera system consists of the cameras itself, motors equipped with encoders rotating the camera and a Raspberry Pi microprocessor board. The Raspberry Pi and the computer of the HMD communicate using a UDP based protocol, [5]. Note that the real camera system/HMD features a stereo camera, which is no loss of generality as the images of both cameras are transferred, taken together and the compensation can be done for both with the proposed idea.

B. Modeling and prediction of the head's movements

We use a rather simple model of the head movements considering only two bio-mechanical degrees of freedoms: the upwards/downwards rotations α^H (upwards $\alpha^H > 0$) and left/right rotations β^H (left $\beta^H > 0$). Additional states are the angular speed of both movements, i.e., the state vector is given by $z = (\alpha^H, \dot{\alpha}^H, \beta^H, \dot{\beta}^H)$. The inputs v to the camera system are the change of the angular speeds of the motion from one time step to another.

We model the resulting reference r_k (the orientation of the HMD) to be tracked by the following linear system[1]

$$z_{k+1} = Fz_k + Gv_k,\ z_k \in \mathbb{Z},\ v_k \in \mathbb{V}, \tag{1a}$$

$$r_k = Hz_k. \tag{1b}$$

Here v_k is the unknown input - the human decisions how to move/accelerate his head. The matrices F and G are

$$F = \begin{pmatrix} F_1 & 0 \\ 0 & F_2 \end{pmatrix}, \qquad F_1 = F_2 = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix},$$

$$G = \begin{pmatrix} G_1 & 0 \\ 0 & G_2 \end{pmatrix}, \qquad G_1 = G_2 = \begin{pmatrix} \frac{T^2}{2} \\ T \end{pmatrix},$$

where T is the HMD's sampling time. The references, i.e., the head orientation, are obtained via (1b) and

$$H = \begin{pmatrix} H_1 & 0 \\ 0 & H_2 \end{pmatrix}, \qquad H_1 = H_2 = \begin{pmatrix} 1 & 0 \end{pmatrix}.$$

Moreover, it is assumed that the rotation, rotational speed as well as the head's acceleration satisfy certain limitations, e.g. the necks maximum turning angle is limited. In detail, we employ constraints where the sets $\mathbb{Z}$ and $\mathbb{V}$, are convex, compact polytopes and contain their origins.

We assume that z_k is measured by the HMD at every time instant. This assumption, i.e., that the head's orientation and velocity (z_k) are fully available, is reasonable as they can be rather precisely measured using e.g. inertial sensors. In contrast, the current and future values of v_k are unknown/uncertain, since the movements of the head are controlled by the human's will. However v_k is such that the constraints are satisfied, i.e., that the human's head rotation, rotational speed as well as acceleration satisfy certain limits.

C. Tracking system - camera system

The camera's dynamics is modeled similar to the head movements: using the upwards/downwards rotation (tilt, α^C) and left/right rotation (pan, β^C) of the camera and their speed results in the state $x = (\alpha^C, \dot{\alpha}^C, \beta^C, \dot{\beta}^C)$. The input u_k corresponds to the applied moments by motors to the camera.

The model is linear, constrained and time-invariant:

$$x_{k+1} = Ax_k + Bu_k,\ x_k \in \mathbb{X},\ u_k \in \mathbb{U}, \tag{2a}$$

$$y_k = Cx_k. \tag{2b}$$

Here $y_k \in \mathbb{R}^2$ corresponds to the viewing direction of the camera (image center). The camera's state and input are subject to constraints, $x_k \in \mathbb{X}$ and $u_k \in \mathbb{U}$, where both sets are convex, compact polytopes and contain their origins.

[1] Linear dynamics are used as this lead to sufficiently good results.

Specifically, we consider as a model for the camera's dynamics two double integrators:

$$A = F, \qquad C = H, \tag{3a}$$

$$B = \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix}, \qquad B_1 = B_2 = c_i \begin{pmatrix} \frac{T^2}{2} \\ T \end{pmatrix}, \tag{3b}$$

where $c_i > 0$ are constants depending on the plant.

Remark 1: (Sampling rate, refresh rate and frame rate) Here, we assumed that the camera and HMD use the same sampling rate, i.e., the frame rate and the refresh rate are equal. In principle, the proposed approach can be extended to the case that the refresh rate of the HMD is larger, which requires to generate intermediate images with the correct pointing direction and field of view.

Remark 2: (Plant model mismatch) Here we assume that the plant model (2) *is exactly known. To consider plant model mismatch the approach needs to be robustifed by allowing disturbances in* (2)*, which is not considered in this work.*

D. Communication channels

We focus mainly on the communication delay: a delay $\delta^I > 0$ from the HMD to the camera and a delay δ^{II} in the reverse direction is assumed, see Figure 1. Thus, the data z_k send at time instant k from the HMD arrives at $k + \delta^I$ at the camera and the images send at k arrive at $k + \delta^{II}$.

E. Overall problem formulation:

We abstract the problem in a general problem setup: one system (the reference generator/HMD) produces a reference output, which the tracking system (camera) should follow, while satisfying constraints. The reference is not exactly known in advance and only available with delay to the tracking system. However the possible changes of the reference are constrained, which should be taken into account.

We aim to design a controller such that the difference of the systems' outputs is below a specific threshold, while satisfying constraints. Mathematically, we aim to guarantee that the tracking error is bounded by a set $\mathbb{D}$:

$$\rho_k = y_k - r_{k+\delta^{II}} \in \mathbb{D}, \tag{4}$$

while satisfying the state constraints $x_k \in \mathbb{X}$ and input constraints $u_k \in \mathbb{U}$. The size of the set $\mathbb{D}$ in our example corresponds to the acceptable tolerance between the field of view of the camera and the field of view of the HMD.

Beyond the desired tracking performance an efficient implementation of the controller should be possible. Moreover, additional objectives such as the reduction of the communication or input utilization are considered, see Section IV.

Remark 3: (Delay compensation approaches) This work considers a tracking task, where the future reference is unknown and also the reference signal is delayed. In contrast, in existing delay compensation approaches such as [16]–[20] only known entities, e.g., the input signals are delayed.

Remark 4: (Non classical tracking) Note that the considered problem is not a classical tracking problem for which a wide range of approaches exist. The reference is not a simple set-point, compare [15], [21], [22], neither a periodic reference as in [23] nor generated by a linear reference model without additive disturbances [24]. Also the reference is not exactly known in advance in contrast to [25]–[27].

III. Main Result

We aim for a simple controller that allows to achieve constraint satisfaction and is real time implementable. The idea is to predict and bound the uncertain future behavior and use these predictions and bounds to determine the input.

A. Basic idea of the approach:

The objective is to control the camera such that the pointing error ρ_k is in the specified set $\mathbb{D}$, see (4), based on the information sent at $k - \delta^I$ from the HMD, while satisfying $x_k \in \mathbb{X}$ and $u_k \in \mathbb{U}$.

The basic idea is to first predict the reference $r_{k+\delta^{II}}$ at the future time instant $k + \delta^{II}$, when the image is used by the HMD, by $\hat{r}_{k+\delta^{II}|k-\delta^I}$ based on $z_{k-\delta^I}$. Second, a controller minimizes the error between y_k and the predicted reference.

To this end, we write the pointing error ρ_k (4) as

$$\rho_k = y_k - r_{\kappa^+} = \underbrace{\hat{r}_{\kappa^+|\kappa^-} - r_{\kappa^+}}_{e_k} + \underbrace{y_k - \hat{r}_{\kappa^+|\kappa^-}}_{t_k}, \tag{5}$$

where $\kappa^+ = k + \delta^{II}$ and $\kappa^- = k - \delta^I$. Note that the first part, the prediction error e_k, depends on the employed prediction approach, whereas the second part, the tracking error t_k depends on the controller design. Both parts, e_k and t_k, are described in the following.

B. Prediction and prediction error

We use a linear model to predict z_{j+m}, $m > 0$ at j by $\hat{z}_{j+m|j}$ based on z_j[2], which is denoted by

$$\hat{z}_{j+m|j} = \mathcal{F}_m z_j, \tag{6}$$

where $\mathcal{F}_m$ is invertible. Here, the prediction uses (1) with zero input ($v_{j+i} = 0$, $i \geq 0$): $\mathcal{F}_m = F^m$, i.e. for the prediction the head's rotation speeds stay the same, see Figure 3. In principle, a different prediction model is possible. For the proposed model, we can bound the prediction error:

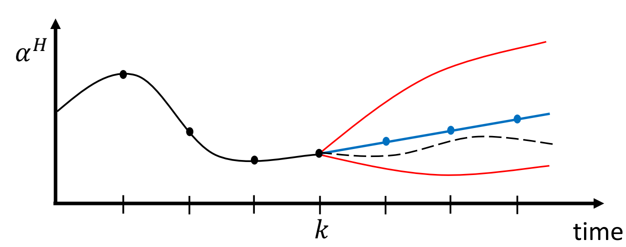

Fig. 3. Illustration: Actual reference (black), reference prediction of (blue) and error bounds (red) for α^H.

Proposition 1: (Bound on worst case prediction error) For any admissible reference $\{r_k\}$ *generated by* (1) *the error*

[2]Here, we use a slightly more general setup to be used in the next section.

between the m step ahead prediction $\hat{r}_{k+m|k}$ and the actual reference value r_{k+m} is bounded by:

$$\hat{r}_{k+m|k} - r_{k+m} \in \mathbb{E}_m(z_{k-m}) \subseteq \overline{\mathbb{E}}_m \tag{7}$$

where

$$\begin{aligned}\mathbb{E}_m(z_{k-m}) = \{\epsilon_m, \epsilon_m = H(\mathcal{F}_m\zeta_0 - \zeta_m) \quad & \quad (8)\\ \text{where } \zeta_k \text{ satisfies for } i = 0, \dots, m-1 &\\ \zeta_{i+1} = F\zeta_i + G\nu_i, &\\ \nu_i \in \mathbb{V},\ \zeta_{i+1} \in \mathbb{Z}, \zeta_0 = z_{k-m}\}.&\end{aligned}$$

and

$$\begin{aligned}\overline{\mathbb{E}}_m = \{\epsilon_m, \epsilon_m = H(\mathcal{F}_m\zeta_0 - \zeta_m) \quad & \quad (9)\\ \text{where } \zeta_k \text{ satisfies for } i = 0, \dots, m-1 &\\ \zeta_{i+1} = F\zeta_i + G\nu_i,\ \nu_i \in \mathbb{V},\ \zeta_{i+1} \in \mathbb{Z}, \zeta_0 \in \mathbb{Z}\}.&\end{aligned}$$

Proof: (Sketch) Any admissible reference sequence $\{r_k\}$ satisfies for $k \geq m$ the conditions of definition of (8), (9) for $i = 0, \dots, m$ and $j = 0, \dots, m-1$ with $z_{k-m+i} = \zeta_i$ and $v_{k-m+j} = \nu_j$. The m step prediction of z_{k-m} is $\mathcal{F}_m z_{k-m}$. Using (5) verifies the above claim. ■

The two sets $\overline{\mathbb{E}}_m$ and $\mathbb{E}_m(z_k)$ are convex polytopes. Note that, $\overline{\mathbb{E}}_m$ is independent of z_k and thus can easily be determined offline. This proposition allows to bound the prediction error e_k, compare (5), by $e_k \in \overline{\mathbb{E}}_{\delta^t}$ or $e_k \in \mathbb{E}_{\delta^t}(z_k)$, where $\delta^t = \delta^I + \delta^{II}$.

Remark 5: (More accurate model) One can use $i > m$ in (8) *to consider that $z_k \in \mathbb{Z}$ holds beyond $k+m$, which provides a more accurate description of the prediction error.*

Remark 6: (Linear prediction model) The linear prediction model (6) *does consider the constraints* (1)*, which would require a nonlinear model. Here, we restrict our attention to such a model, since it allows an efficient computation of the prediction and of the required worst case bounds* (8)*.*

C. Baseline tracking controller and tracking error

The idea to minimize the tacking error is to design a linear controller, which determines u_k based on x_k and z_{κ^-}.

1) Error dynamics: In order to describe the tracking error t_k, we combine the head movements/reference model (1) and the camera system (2). For $k \geq \delta^I$ we have

$$\begin{pmatrix} x_{k+1} \\ \hat{z}_{\kappa^+ +1|\kappa^- +1}\end{pmatrix} = \mathcal{A}\begin{pmatrix} x_k \\ \hat{z}_{\kappa^+|\kappa^-}\end{pmatrix} + \mathcal{B}\begin{pmatrix} u_k \\ v_{\kappa^-}\end{pmatrix} \tag{10a}$$

$$\text{where } \mathcal{F}_{\delta^t}^{-1}\hat{z}_{\kappa^+|\kappa^-} \in \mathbb{Z},\ v_{\kappa^-} \in \mathbb{V}, \tag{10b}$$

$$\mathcal{A} = \begin{pmatrix} A & 0 \\ 0 & \mathcal{F}_{\delta^t}F\mathcal{F}_{\delta^t}^{-1}\end{pmatrix} \qquad \mathcal{B} = \begin{pmatrix} B & 0 \\ 0 & \mathcal{F}_{\delta^t}G\end{pmatrix} \tag{10c}$$

with $\delta^t = \delta^I + \delta^{II}$ and the tracking error t_k as output

$$t_k = \begin{pmatrix} C & -H\end{pmatrix}\begin{pmatrix} x_k \\ \hat{z}_{\kappa^+|\kappa^-}\end{pmatrix}. \tag{10d}$$

Note that in (10) we utilized

$$\begin{aligned}\hat{z}_{\kappa^+ +1|\kappa^- +1} &= \mathcal{F}_{\delta^t} z_{\kappa^- +1} = \mathcal{F}_{\delta^t}(F z_{\kappa^-} + G v_{\kappa^-})\\ &= \mathcal{F}_{\delta^t}F\mathcal{F}_{\delta^t}^{-1}\hat{z}_{\kappa^+|\kappa^-} + \mathcal{F}_{\delta^t}G v_{\kappa^-}.\end{aligned}$$

Introducing $a_k = x_k - \hat{z}_{\kappa^+|\kappa^-}$ and using $C = H$ we obtain

$$a_{k+1} = (A - \mathcal{F}_{\delta^t}F\mathcal{F}_{\delta^t}^{-1})\hat{z}_{\kappa^+|\kappa^-} + Aa_k - \mathcal{F}_{\delta^t}Gv_{\kappa^-} + Bu_k,$$

which simplifies to

$$a_{k+1} = Aa_k + Bu_k - F^{\delta^t}Gv_{\kappa^-}, \tag{11a}$$

$$t_k = C\alpha_k, \tag{11b}$$

since we assume $A = F$ and $\mathcal{F}_{\delta^t} = F^{\delta^t}$.

2) Control law and bounding the tracking error: We focus on the utilization of a linear controller for (11):

$$u_k = Ka_k = K(x_k - \hat{z}_{\kappa^+|\kappa^-}), \tag{12}$$

where the controller gain K is designed such that $A+BK$ is asymptotic stable. Moreover, the choice of K should aim to minimize t_k, while satisfying the state and input constraints as explained lateron. Note that asymptotic stability and boundedness of v_{κ^-} by $\mathbb{V}$ guarantees that for the dynamics (11), (12), a robust positive invariant set exists.

Remark 7: (Nonlinear control laws) We propose to use a linear control law to allow an efficient implementation of the controller. An alternative is to use closed loop minmax MPC, which would be computationally more challenging to implement, see e.g. [15], [30], [31].

3) Overall result: We combine the above results to obtain conditions guaranteeing that the pointing error ρ_k satisfies the bound $\rho_k \in \mathbb{D}$ and that the constraints are satisfied. Therefore, we assume for the sets $\mathbb{A}$ and $\mathbb{E}_{\delta^t}$:

Assumption 1: (Requirements on sets $\mathbb{A}$, $\mathbb{E}_{\delta^t}$) The sets $\mathbb{A}$ and $\mathbb{E}_{\delta^t}$ satisfy:

- *Robust positive invariance: If $a_k \in \mathbb{A}$, then $a_{k+1} \in \mathbb{A}$, for any $v_{\kappa^-} \in \mathbb{V}$ and a_{k+1} given by* (11), (12)*,*
- *Constraint satisfaction: $K\mathbb{A} \subseteq \mathbb{U}$, $\mathbb{A} \subseteq \mathbb{X} \ominus \mathcal{F}_{\delta^t}\mathbb{Z}$,*
- *Error bound satisfaction: $C\mathbb{A} \oplus \mathbb{E}_{\delta^t} \subseteq \mathbb{D}$.*

This assumption allows us to state the main result:

Corollary 1: (Guaranteed tracking) Let Assumption 1 hold. If $a_\delta^I \in \mathbb{A}$, then the closed loop system (1)*,* (2)*,* (6) *and* (12) *guarantees $u_k \in \mathbb{U}$, $x_k \in \mathbb{X}$ and $\rho_k \in \mathbb{D}$* (4)*.*

Proof: (Sketch) First $a_\delta^I \in \mathbb{A}$ guarantees $a_k^I \in \mathbb{A}$, $k \geq \delta^I$ due to the robust positive invariance of $\mathbb{A}$ (part of Assumption 1). The first claim, the constraint satisfaction, follows from the second part of of Assumption 1 and (for the state constraints) using $x_k = a_k + z_{\kappa^+|\kappa^-}$ together with the constraint on z. The last part (error bound pointing error) follows from the decomposition of the error into prediction and tracking error (5) and the bounds $\mathbb{E}_{\delta^t}/C\mathbb{A}$ on the prediction error and tracking error $t_k = Ca_k$, respectively. ■

In a nutshell, Corollary 1 together with Assumption 1 provides a relationship between the prediction error and tracking error and the threshold set $\mathbb{D}$, i.e., the required differences between the field of views. Moreover, since the delay has an influence on the prediction and tracking error, also the influence of the delay can be easily seen.

Note that $a_\delta^I \in \mathbb{A}$ is required in Corollary 1 as for $k < \delta^I$, no predicted reference $\hat{z}_{\kappa^+|\kappa^-}$ is available.

Remark 8: (Set computation) For the computation of the set $\mathbb{E}_{\delta^t}$ and $\mathbb{A}$ tools such as the MPT3 [32] can be used.

IV. Event-triggered methods

This section outlines extensions based on event-triggered control [6]. We present approaches to decrease the network load by avoiding unnecessary image transmissions and to reduce actuator utilizations The ideas can be combined.

A. Avoiding image transmissions

In the above scheme, at every time the images are sent from the camera to the HMD, which leads to a high network load. To reduce the network load, we propose an event-triggered transmission scheme for (almost) static scenes, which avoids to send unnecessary images. We use γ_k to denote for $k > \delta^I$ the last sent image:

$$\gamma_{k+1} = \begin{cases} k, & \text{if at } k \text{ an image is sent} \\ \gamma_k, & \text{otherwise.} \end{cases} \tag{13}$$

At $k = \delta^I$ (first reference available at camera) always an image is sent. The proposed event-triggered transmission scheme should only send an image to the HMD, if the last sent image cannot be reused. In detail, no new image needs to be sent if the last image sent contains also the image to be sent at the current time instant, i.e. it needs to hold that

$$\omega_k = y_{\gamma_k} - r_{\kappa_+} \in \mathbb{D}. \tag{14}$$

A simple condition guaranteeing that (14) holds is:

Corollary 2: (Event-triggered image transmission) Let Assumption 1 and $a_\delta^I \in \mathbb{A}$ hold. If the triggering condition

$$y_{\gamma_k} - \hat{r}_{\kappa_+|\kappa_-} \in \mathbb{D} \ominus \mathbb{E}_{\delta^t} \tag{15}$$

holds, then no image needs to be transmitted, i.e., $\gamma_{k+1} = \gamma_k$.

Proof: (Sketch) The prediction error is independent of the image transmission, i.e. $\hat{r}_{\kappa_+|\kappa_-} - r_{\kappa_+} \in \mathbb{E}_{\delta^t}$. Therefore, if the triggering condition (15) holds (14) is satisfied. ■

The set $\mathbb{D} \ominus \mathbb{E}_{\delta^t}$ can be computed offline, which allows to check the triggering condition (15) efficiently.

One can compute online tighter error bounds using the sets $\mathbb{E}_\delta^t(z_{k-\delta^t})$, see Proposition 1, which is computationally more expensive, but can reduce the amount of the transmissions.

B. Avoiding actuator utilizations

In the following, we present an event-triggered approach, which allows to use a zero input ($u_k = 0$) instead of the input u_k given by the control law (12), while guaranteeing the bound on the pointing error (4) and satisfaction of the constraints $x_k \in \mathbb{X}$ and $u_k \in \mathbb{U}$. Using a zero input is beneficial in some applications, e.g., to reduce the wear and tear of the actuator or the energy consumption, see [4], [6].

Corollary 3: (Event-triggered control inputs) Let Assumption 1 hold. If $a_k \in \mathbb{A}$ and the triggering condition

$$Ax_k \in \mathbb{X} \tag{16a}$$

$$Ax_k - \hat{z}_{\kappa_+ +1|\kappa_-} \in \mathbb{A} \ominus \mathcal{F}_{\delta^t} G \mathbb{V} \tag{16b}$$

holds, then using $u_k = 0$ is possible, i.e. $u_{k+i} \in \mathbb{U}$, $x_{k+i} \in \mathbb{X}$ and $\rho_{k+i+1} \in \mathbb{D}$, $i \geq 0$ can be guaranteed.

Proof: First let us verify the constraint satisfaction. We have $u_k = 0 \in \mathbb{U}$, compare Section II. Moreover, (16a) guarantees that $x_{k+1} \in \mathbb{X}$.

To guarantee satisfaction of the input/state constraints beyond $k/k+1$ and that the bound on the tracking error ρ is valid we rely on the basic scheme (Section II). We need to verify that $a_{k+1} \in \mathbb{A}$ holds for $u_k = 0$ and for all possible $v_{\kappa^-} \in \mathbb{V}$ such that the basic scheme (Section III) can be used for $k+1$. Using $\hat{z}_{\kappa_+ +1|\kappa_- +1} - \hat{z}_{\kappa_+ +1|\kappa_-} = \mathcal{F}_{\delta^t} G v_{\kappa^-}$, one can see that the choice of the set in (16a) guarantees this. ■

The condition (16) can be efficiently evaluated if the set in (16b) is computed offline.

V. Simulation example

We outline the proposed approach by simulations. The sampling time is chosen as $T = 0.011ms$ (the refresh rate of the display) and the actuator constants are $c_1 = c_2 = 1$. The constraints on the heads movements are assumed to be

$$\mathbb{Z} = \left\{ z \text{ s.t. } \begin{pmatrix} -30 \\ -60 \\ -50 \\ -100 \end{pmatrix} \leq z \leq \begin{pmatrix} 30 \\ 60 \\ 50 \\ 100 \end{pmatrix} \right\},$$
$$\mathbb{V} = \left\{ v \text{ s.t. } \begin{pmatrix} -90 \\ -150 \end{pmatrix} \leq v \leq \begin{pmatrix} 90 \\ 150 \end{pmatrix} \right\}.$$

The constraints of the camera system (2) are

$$\mathbb{X} = \left\{ x \text{ s.t. } \begin{pmatrix} -40 \\ -80 \\ -60 \\ -120 \end{pmatrix} \leq x \leq \begin{pmatrix} 40 \\ 80 \\ 60 \\ 120 \end{pmatrix} \right\},$$
$$\mathbb{U} = \left\{ u \text{ s.t. } \begin{pmatrix} -180 \\ -280 \end{pmatrix} \leq u \leq \begin{pmatrix} 180 \\ 280 \end{pmatrix} \right\}.$$

A. Bound on prediction error

Due to the system structure (no coupling between α and β coordinates) the bounds $\overline{\mathbb{E}}_m$ (Proposition 1) on the prediction error $\hat{r}_{k+m|k} - r_{k+m}$ take the form

$$\overline{\mathbb{E}}_m = \left\{ e \text{ s.t. } \begin{pmatrix} -\bar{e}_m^\alpha \\ -\bar{e}_m^\beta \end{pmatrix} \leq e \leq \begin{pmatrix} \bar{e}_m^\alpha \\ \bar{e}_m^\beta \end{pmatrix} \right\}. \tag{17}$$

Figure 4 illustrates these bounds as well as the errors bounds if no predictions are used ($\mathcal{F}_m = I$). As expected the error bounds grow for increasing delay and without prediction the bounds are significantly larger.

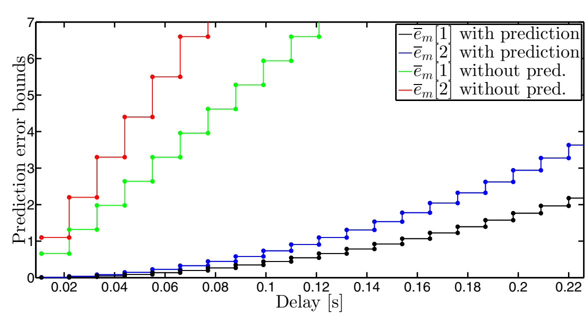

Fig. 4. Bounds on prediction error $\overline{\mathbb{E}}_m$ (17) with/without prediction.

B. Time-triggered control

We assume that a total delay of $\delta^t = 6$ time units (66ms) is present ($\delta^I = \delta^{II} = 3$) and that the tolerance bound $\mathbb{D}$ is

$$\mathbb{D} = \left\{ u \text{ s.t. } \left(\begin{smallmatrix} -2 \\ -3.25 \end{smallmatrix}\right) \le u \le \left(\begin{smallmatrix} 2 \\ 3.25 \end{smallmatrix}\right) \right\}.$$

Using as gain K for the tracking controller (12)

$$K = \left(\begin{smallmatrix} -128.9256 & -22.0182 & 0 & 0 \\ 0 & 0 & -128.9256 & -22.0182 \end{smallmatrix}\right),$$

allows to compute $\mathbb{A}$ satisfying together with $\mathbb{E}_{\delta^t}$ from above Assumption 1, i.e. the scheme of Section III is applicable.

Figures 5-7 illustrate the proposed approach (Section III)[3]. We see that the constraints are satisfied and that the remaining tracking error is within the desired bounds.

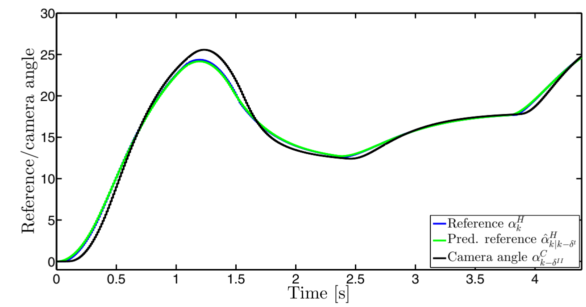

Fig. 5. Time triggered control: Reference, camera angle (α component)

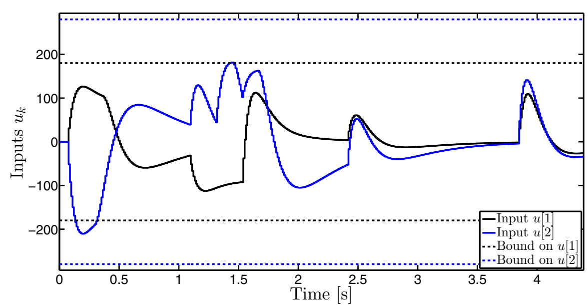

Fig. 6. Time triggered control: Inputs u_k and input constraints $\mathbb{U}$.

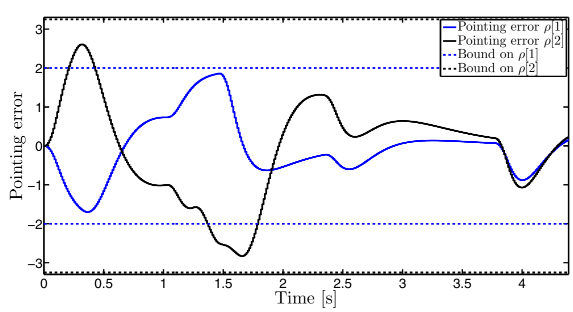

Fig. 7. Time triggered control: Pointing error.

[3]For the illustration the time is "corrected", i.e. the delay is removed.

C. Event-triggered image transmission

Figures 8, 9 outline the approach using the event-triggered image transmissions scheme from Section IV-A. Note that the camera positions/inputs are the same as above and therefore not shown again. We observe that the bound on the pointing error ω_k (14) is always satisfied. Note that the proposed approach avoids around 79% of the transmissions and therefore enables a reduction of the network load.

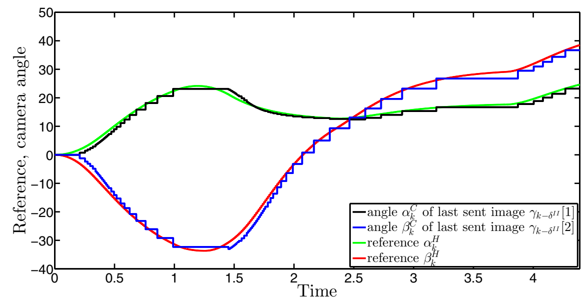

Fig. 8. Event-triggered image transmission: reference, camera angles at last sent image.

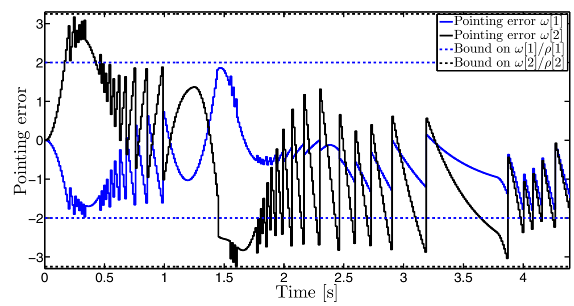

Fig. 9. Event-triggered image transmission: Pointing error ω_k (14) .

D. Event-triggered control

The proposed event-triggered control approach (Section IV-B) is illustrated in Figure 10 and Figure 11. Observe that the applied input is often zero as desired: at 61% of the time instants zero input $u_k = 0$ is applied. Moreover, one can verify that the constraints on the input, state and error (not shown here).

VI. Summary and Future work

This work considered the control of an actuated camera via a head mounted display and a wireless communication network by a human. The objective is to control the camera such that the viewing angle of the human and the image displayed match each other, even so the communication is subject to delays. Therefore, a control scheme and design approach based on set theoretic methods was proposed. In detail, the controller guarantees that the error between the camera and the predicted viewing direction is within specified bounds. This allows to compensate the remaining pointing error via simple image processing methods, if for the displayed image a field of view smaller than the one of the camera is used. We derived conditions determining the required difference of

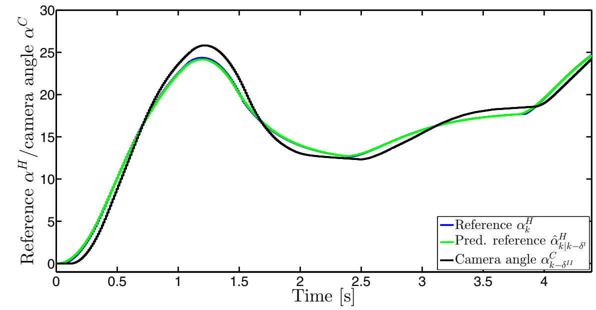

Fig. 10. Event-triggered control: Reference, camera angle (α component).

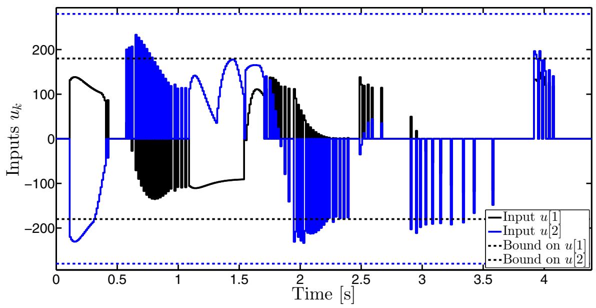

Fig. 11. Event-triggered control: Inputs u_k and input constraints $\mathbb{U}$.

the fields of view. The approach was extended using event-triggered ideas such that images are only sent to the display if necessary. Simulations illustrated the proposed control approach, its applicability to the proposed control task and the potential of event-triggered control to reduce the network load and actuator utilizations.

Future working directions include the extension of the proposed approach and its detailed evaluation on the real system.

References

[1] K. Johansson, G. Pappas, P. Tabuada, and C. Tomlin, "Special issue on control of cyber-physical systems," *IEEE Trans. Automatic Control*, vol. 59, pp. 3120–3121, 2014.

[2] R. A. Gupta and M.-Y. Chow, "Networked control system: Overview and research trends," *IEEE Trans. Industrial Electronics*, vol. 57, no. 7, pp. 2527–2535, 2010.

[3] J. Lunze, *Control theory of digitally networked dynamic systems*. Springer, 2014.

[4] S. Lucia, M. Kögel, P. Zometa, D. E. Quevedo, and R. Findeisen, "Predictive control, embedded cyberphysical systems and systems of systems a perspective," *Annual Reviews in Control*, vol. 41, pp. 193 – 207, 2016.

[5] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. Prentice Hall, 2010.

[6] W. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. IEEE Conference on Decision and Control*, 2012, pp. 3270–3285.

[7] D. E. Quevedo, P. K. Mishra, R. Findeisen, and D. Chatterjee, "A stochastic model predictive controller for systems with unreliable communications," in *Proc. IFAC Conf. NMPC*, 2015, pp. 57–64.

[8] B. D. Lawson, *Motion Sickness Symptomatology and Origins*. CRC press, 2014, pp. 531–599.

[9] M. Yasuda and K. Kawakami, "New method of monitoring remote wildlife via the internet," *Ecological Research*, vol. 17, no. 1, pp. 119–124, 2002.

[10] R. A. Long, P. MacKay, J. Ray, and W. Zielinski, *Noninvasive survey methods for carnivores*. Island Press, 2012.

[11] T. D. Räty, "Survey on contemporary remote surveillance systems for public safety," *IEEE Trans. Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 5, pp. 493–515, 2010.

[12] A. Kiruluta, M. Eizenman, and S. Pasupathy, "Predictive head movement tracking using a Kalman filter," *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 27, no. 2, pp. 326–331, 1997.

[13] H. Himberg and Y. Motai, "Head orientation prediction: delta quaternions versus quaternions," *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1382–1392, 2009.

[14] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[15] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2012.

[16] A. Bemporad, "Predictive control of teleoperated constrained systems with unbounded communication delays," in *Proc. IEEE Conf. Decision and Control*, vol. 2, 1998, pp. 2133–2138.

[17] P. Varutti, B. Kern, T. Faulwasser, and R. Findeisen, "Event-based model predictive control for networked control systems," in *Proc. IEEE Conf. Decision and Control and Chinese Control Conf.*, 2009, pp. 567–572.

[18] G. Pin and T. Parisini, "Networked predictive control of uncertain constrained nonlinear systems: Recursive feasibility and input-to-state stability analysis," *IEEE Trans. Automatic Control*, vol. 56, no. 1, pp. 72–87, 2011.

[19] R. Findeisen and P. Varutti, "Stabilizing nonlinear predictive control over nondeterministic communication networks," in *Nonlinear model predictive control*. Springer, 2009, pp. 167–179.

[20] R. Findeisen and F. Allgöwer, "Computational delay in nonlinear model predictive control," in *Proc. Int. Symp. Adv. Control of Chemical Processes*, 2003, pp. 427–432.

[21] D. Limón, I. Alvarado, T. Alamo, and E. F. Camacho, "MPC for tracking piecewise constant references for constrained linear systems," *Automatica*, vol. 44, no. 9, pp. 2382–2387, 2008.

[22] D. Simon, J. Löfberg, and T. Glad, "Reference tracking MPC using dynamic terminal set transformation," *IEEE Trans. Automatic Control*, vol. 59, no. 10, pp. 2790–2795, 2014.

[23] D. Limon, M. Pereira, D. M. de la Pena, T. Alamo, C. N. Jones, and M. N. Zeilinger, "MPC for Tracking Periodic References," *IEEE Trans. on Automatic Control*, vol. 61, no. 4, pp. 1123–1128, 2016.

[24] U. Maeder and M. Morari, "Offset-free reference tracking with model predictive control," *Automatica*, vol. 46, no. 9, pp. 1469–1476, 2010.

[25] M. Kögel and R. Findeisen, "On MPC based trajectory tracking," in *Proc. European Control Conf.*, 2014, pp. 121–127.

[26] T. Faulwasser and R. Findeisen, "A model predictive control approach to trajectory tracking problems via time-varying level sets of Lyapunov functions," in *Proc. IEEE Conf. Decision & Control*, 2011, pp. 3381–3386.

[27] T. Faulwasser and R. Findeisen, "Nonlinear model predictive control for constrained output path following," *IEEE Trans. Automatic Control*, vol. 61, no. 4, pp. 1026–1039, 2016.

[28] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[29] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer, 2008.

[30] J. Löfberg, *Minimax approaches to robust model predictive control*. Linköping University Electronic Press, 2003, vol. 812.

[31] P. O. Scokaert and D. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Trans. Automatic control*, vol. 43, no. 8, pp. 1136–1142, 1998.

[32] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. European Control Conf.*, 2013, pp. 502–510, http://control.ee.ethz.ch/ mpt.