Robot Navigation Under Uncertainties Using Event Based Sampling

Michele Colledanchise,[†] Dimos V. Dimarogonas,[‡] and Petter Ögren[†]

***Abstract*— In many robot applications, sensor feedback is needed to reduce uncertainties in environment models. However, sensor data acquisition also induces costs in terms of the time elapsed to make the observations and the computations needed to find new estimates. In this paper, we show how to use event based sampling to reduce the number of measurements done, thereby saving time, computational resources and power, without jeopardizing critical system properties such as safety and goal convergence. This is done by combining recent advances in nonlinear estimation with event based control using artificial potential fields. The results are particularly useful for real time systems such as high speed vehicles or teleoperated robots, where the cost of taking measurements is even higher, in terms of stops or transmission times. We conclude the paper with a set of simulations to illustrate the effectiveness of the approach and compare it with a baseline approach using periodic measurments.**

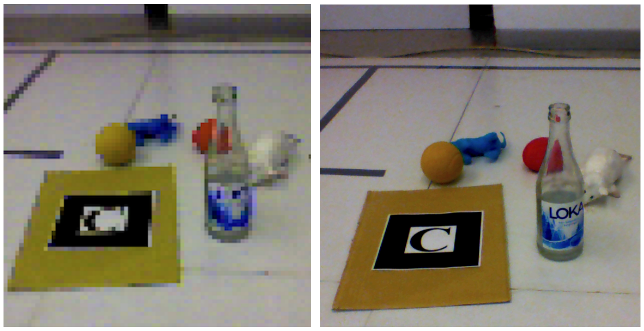


Fig. 1. Streamed data from a humanoid NAO robot during gait (left) and transmitted data while it stands still (right). As can be seen, stopping is often necessary to achieve good sensor data. Thus the controller needs to make a tradeoff between saving time and reducing uncertainty.

I. Introduction

The control and sensor systems in a robot are often designed separately. The sensor system is designed to deliver information with low uncertainty and high frequency, while the control system is designed to achieve some control objectives, preferably taking the sensor system specifications into account. This separation is often practical, in terms of system design, but can also have a negative impact on overall system performance. In particular, there are cases where there is a significant cost associated with sensing. If the sensor is active, such as a radar, a sonar, or a laser scanner, power consumption can be a significant cost (especially for small, long endurance vehicles). If the computations associated with sensing are significant and share the same processing unit with other critical functions (e.g., planning), the performance of those might improve if sensing computations are reduced. Finally, if particular actions are needed to enable sensing, such as stopping, see Fig. 1, mission times might decrease when sensing is reduced. These improvements are even more significant in high speed applications where stopping takes time, or in teleoperated applications where transmission times add to the seconds lost.

The benefits of designing the combined control and sensing policy can be seen by the following example. If a robot is far away from an obstacle, the exact location of that obstacle is often not needed. Furthermore, the uncertainties associated with the computed estimates often increase with distance, so any measurement taken is bound to have large uncertainties. Therefore measurement intervals can be sparse. On the other hand, if the robot is moving in a cluttered environment, close to obstacles, lower uncertainties are needed, and can also be supplied by the sensor system. Thus, fairly short measurement intervals might be sufficient to guarantee collision avoidance. In this paper, we formalize the situation described above and compute sensor triggering conditions to satisfy the real information need of the controller.

The contribution of this paper is the combination of the recent advances on estimation for nonlinear systems, the provable collision-free path planning provided by an artificial potential field and the low computational cost of the event based control. Doing this, we are able to drastically reduce the computational demand of the navigation, without jeopardizing critical requirements such as collision avoidance and destination convergence.

The outline of the paper is as follows: In Section II, we discuss the related work. In Section III we present the problem formulation. Section IV then describes the proposed solution. The theoretical properties of the solution are analyzed in Section V. Finally, simulations are presented to illustrate the approach in Section VI and conclusions are drawn in Section VII.

II. Related work

A. Path Planning

Robot navigation has received a lot of attention in the last decades [1], [2] due to its crucial role in various tasks as exploration, search and rescue, surveillance, coverage, cooperative manipulation, etc. Navigation can either be done giving a strict sequence of movements, so-called *open loop execution*, or using continuous feedback to provide robust-

[†] The author is with the Centre for Autonomous Systems, Computer Vision and Active Perception Lab, School of Computer Science and Communication, The Royal Institute of Technology - KTH, Stockholm, Sweden.

[‡] The author is with the Centre for Autonomous Systems, Automatic Control Lab, School of Electrical Engineering, The Royal Institute of Technology - KTH, Stockholm, Sweden.

ness in the presence of disturbances and noise, so-called *closed loop execution*.

Artificial potential fields are a class of methods introduced by Khatib [3] to provide robust *closed loop execution* of autonomous navigation. By defining a potential field in the robot's workspace, and then following the negative gradient of the potential field, destination convergence is achieved. However, the artificial potential field approach can suffer from the presence of local minima, locations where the robot can get stuck. Therefore, Rimon and Koditschek [4] extended the potential field approach by demonstrating that their Navigation Functions (NFs), under some assumptions, give potentials fields free of local minima. Recently, NFs found use in different scenarios, including limited sensing capabilities of non-holonomic systems [5], multi-agent systems [6], [7], [8], [9], different obstacle shapes [10] and linear temporal logic specifications [11]. In this paper we use NFs to provide a robust closed loop execution for the problem at hand. However, our approach is different from the above related work in that it is formulated in discrete time. This enables us to incorporate the event based sampling, and also avoids the well known problem of having some initial positions from where the desired destination cannot reached.

B. Event Based Control

The core idea of Event Driven (ED) control is to replace the periodic sensing and action with a more elaborate timing choice [12], [13], [14], [15]. ED has found applications in different field, such as industrial control systems [16], [17], network control systems [18], power networks [19], sensor networks [20], [21], estimation and optimization [22], control of stochastic systems [23], and autonomous navigation [24]. Later works show how the specified performance of the overall control system is maintained after the introduction of a ED framework [25], [26], and different comparisons between time driven and ED control, both highlighting a better real-time performance of ED, are found in [27], [28].

Finally, experimental evidence supports that ED control improves the control performances and appears to drastically reduce the communication and computational demand required by real-time systems [29], [30], [31].

As stated above, in this paper we combine an ED approach [29] with NFs [4] and new results on nonlinear estimation [32] to address the problem of doing navigation under uncertainties. To the best of our knowledge, this has not been done before.

III. Problem Formulation

Consider a system of a robot moving among $N \in \mathbb{N}$ obstacles operating in a workspace $\mathcal{W} \subset \mathbb{R}^n$ with $\mathcal{W} = \{\tilde{q} \in \mathbb{R}^n : ||\tilde{q} - o_0|| \leq \rho_0 \}$ and o_0 being the center of the workspace and ρ_0 its radius. Let $q \in \mathcal{W}$ denote the center position of the robot and $r \in \mathbb{R}_+$ its radius, $o_i \in \mathcal{W}$ the center position of the obstacle i and $\rho_i \in \mathbb{R}_+$ its radius. Furthermore, let $d \in \mathcal{W}$ be the robot destination and $B_d \in \mathbb{R}_+$ a margin such that the destination is successfully reached if $||q - d|| \leq B_d$. The robot dynamics is described by the discrete-time equations:

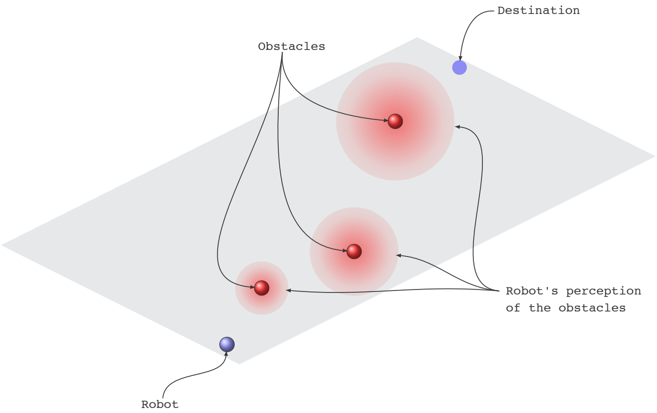


Fig. 2. Example Scenario

$$q(k+1) = f(q(k)) + Bu(q(k)) + v(k) \quad (1)$$
$$y_a(k) = q(k) + \omega_a(k) \quad (2)$$

where $u : \mathbb{R}^n \to \mathbb{R}^p$ is the control input, B is a $n \times p$ matrix, $f : \mathcal{W} \to \mathcal{W}$ is a Lipschitz continuous function with L_f as the Lipschitz constant, $v(k) \in \mathbb{R}^n$ is an additive disturbance affecting the robot dynamics and ω_a is an additive noise affecting $y_a(k)$, which is the robot's measurements of its own position. The dynamics of the i-th obstacle is described by the discrete-time system:

$$o_i(k+1) = g_i(o_i(k)) \quad (3)$$
$$y_{oi}(k) = o_i(k) + \omega_i(k) \quad (4)$$
$$y_{\rho_i}(k) = \rho_i + \psi_i(k) \quad (5)$$

where $g_i : \mathcal{W} \to \mathcal{W}$ is a Lipschitz continuous function with L_g as the Lipschitz constant. To be less restrictive $g_i(\cdot)$ can be either known (e.g., for static obstacles $g_i(o_i(k)) = o_i(k)$) or approximated by a function $\tilde{g}_i(\cdot)$ with $\tilde{L}_g$ as an upper bound of the Lipschitz constant such that $L_g \leq \tilde{L}_g$. Moreover, $y_{oi}(k)$ is the measured position of the i-th obstacle and $y_{\rho i}(k)$ is the measured i-th obstacle's radius at time k. ω_i and ψ_i with $i = \{1, 2, \ldots, N\}$ are additive measurement noises. The disturbances and noise are unknown with given upper bounds: $||\omega_a(k)|| \leq \bar{\omega}_a$; $||\omega_i(k)|| \leq \bar{\omega}_i$; $||\psi_i(k)|| \leq \bar{\psi}_i$; and $||v(k)|| \leq \bar{v} \ \forall k$. Note that the upper bounds are fixed for clarity of exposition. However, the formulation can be easily extended to distance-varying bounds, representing the fact that the uncertainties on the obstacles' position vary with their distance with respect to the robot, as depicted in Fig. 2.

Problem 1: Given a system described as above, define a control input $u(q(k))$ based on the last measurements at discrete time τ (i.e., $y_a(\tau)$, $y_{oi}(\tau)$, and $y_{\rho i}(\tau)$) with $k \geq \tau$ such that:

- at any time k collision avoidance is guaranteed, i.e.
$$||q(k+1) - o_i(k+1)|| > (r + \rho_i(k)) \ \forall k.$$
- at some time T the destination is successfully reached, i.e.
$$||q(k) - d|| \leq B_d \ \forall k \geq T.$$

IV. Proposed Solution

We use discrete control inputs $u(q(k))$ to satisfy Problem 1. The control inputs are computed upon the robot and obstacles positions, estimated at each time step based on the last measurement taken at time step τ. At each time step k two so called *triggering conditions* are evaluated to establish if a new measurement is needed.

A. Filtering and prediction

Since the statistics of the disturbance and noises are assumed to be unknown, a useful criterion to estimate[1] $q(k)$, $o_i(k)$, and $\rho_i(k)$ $\forall i = \{0,1,\dots,N\}$ consists in a least-squares approach. Unfortunately, the exact optimal estimation requires a large computational effort and the minimization of a *full information cost* [33]. A reasonable approximation is given by using the *ε-optimal moving horizon estimator* [32] which provides an upper bound for the estimation error given a optimality gap $\varepsilon > 0$ as follows:

$$\|q(k) - \hat{q}(k|k)\| \leq \sqrt{\xi_q(k,\varepsilon)} \tag{6}$$

$$\|o_i(k) - \hat{o}_i(k|k)\| \leq \sqrt{\xi_{oi}(k,\varepsilon)} \ \forall i = 0,1,\dots,N \tag{7}$$

$$|\rho_i(k) - \hat{\rho}_i(k|k)| \leq \sqrt{\xi_{\rho_i}(k,\varepsilon)} \ \forall i = 0,1,\dots,N \tag{8}$$

Further details about the derivation of the functions in the right hand side of (6)-(8) can be found in [32]. In the next section we present the choice of ε.

Remark 1: In many practical cases the above upper bounds can also be derived experimentally or found in the sensors' characterization given by the manufacturer.

Assuming that τ is the latest measurement time step and k is the current time step, when the measurements $y_a(\tau)$, $y_{oi}(\tau)$, and $y_{\rho i}(\tau)$ are used to estimate $q(k)$, $o_i(k)$, and $\rho_i(k)$ $\forall i = 0,1,\dots,N$ with $k \geq \tau$, then $\hat{q}(k+1|\tau)$, $\hat{o}_i(k+1|\tau)$, and $\hat{\rho}_i(k+1|\tau)$ are called *predictions*.

Lemma 1: Let $B_q : \mathbb{N} \times \mathbb{N} \to \mathbb{R}_+$ be a function defined as:

$$B_q(k,\tau) = \begin{cases} L_f^{k-\tau}\sqrt{\xi_q(\tau,\varepsilon)} + \sum_{i=0}^{k-\tau-1}(L_f^i\bar{v}) + & \\ + \sum_{i=0}^{k-\tau-1} L_f^i \|B\|\bar{u}(k-1-i)) & \text{if } k > \tau \\ \sqrt{\xi_q(k,\varepsilon)} & \text{if } k = \tau \\ 0 & \text{if } k < \tau \end{cases} \tag{9}$$

where $\bar{u} : \mathbb{N} \to \mathbb{R}_+$ such that $\bar{u}(k) \geq \|u(q(k)) - u(\hat{q}(k|\tau))\|$. Then $B_q(k+1,\tau)$ gives an upper bound of the prediction error of the robot position $\forall k \geq \tau$:

$$\|q(k+1) - \hat{q}(k+1|\tau)\| \leq B_q(k+1,\tau). \tag{10}$$

Proof: From (2) we have:

$$\begin{aligned} &\|q(k+1) - \hat{q}(k+1|\tau)\| = \\ &= \|f(q(k)) + Bu(q(k)) + v(k) - f(\hat{q}(k|\tau)) - Bu(\hat{q}(k|\tau))\| \leq \\ &\leq \|f(q(k)) - f(\hat{q}(k|\tau))\| + \|Bu(q(k)) - Bu(\hat{q}(k|\tau))\| + \bar{v} = \\ &= \|f(f(q(k-1)) + Bu(q(k-1)) + v(k-1)) \\ &- f(f(\hat{q}(k-1|\tau)) - Bu(\hat{q}(k-1|\tau)))\| + \|B\|\bar{u}(k) + \bar{v} \end{aligned} \tag{11}$$

from Lipschitz continuity

$$\begin{aligned} &\|f(f(q(k-1)) + Bu(q(k-1)) + v(k-1)) \\ &- f(f(\hat{q}(k-1|\tau)) - Bu(\hat{q}(k-1|\tau)))\| + \\ &+ \|Bu(q(k)) - Bu(\hat{q}(k|\tau))\| + \bar{v} \leq \\ &\leq L_f\|f(q(k-1)) + Bu(q(k-1)) - f(\hat{q}(k-1|\tau)) + \\ &- Bu(\hat{q}(k-1|\tau))\| + L_f\bar{v} + \bar{v} + \\ &+ \|Bu(q(k)) - Bu(\hat{q}(k|\tau))\| \leq \\ &\leq L_f\|f(q(k-1)) + Bu(q(k-1)) - f(\hat{q}(k-1|\tau)) + \\ &- Bu(\hat{q}(k-1|\tau))\| + L_f\bar{v} + \bar{v} + \\ &+ \|Bu(q(k)) - Bu(\hat{q}(k|\tau))\| + \\ &+ L_f\|Bu(q(k-1)) - Bu(\hat{q}(k-1|\tau))\| \end{aligned} \tag{12}$$

continuing the Lipschitz argument until we get in the right hand side $\|q(\tau) - \hat{q}(\tau|\tau)\|$, for which we have the bound $\sqrt{\xi_q(\tau,\varepsilon)}$, the following holds:

$$\begin{aligned} &\|q(k+1) - \hat{q}(k+1|\tau)\| \leq L_f^{k-\tau+1}\|q(\tau) - \hat{q}(\tau|\tau)\| + \\ &+ (1 + L_f + L_f^2 + \ldots + L_f^{k-\tau-1})\bar{v} + \\ &+ \|B\|\bar{u}(k-1) + \ldots + L_f^{k-\tau-1}\|B\|\bar{u}(\tau) = B_q(k+1,\tau) \end{aligned} \tag{13}$$

■

Lemma 2: Let $B_{oi} : \mathbb{N} \times \mathbb{N} \to \mathbb{R}_+$ be a function defined as:

$$B_{oi}(k,\tau) \triangleq \tilde{L}_g^{k-\tau}\sqrt{\xi_{oi}(\tau,\varepsilon)}. \tag{14}$$

Then $B_{oi}(k+1,\tau)$ gives upper bound of the prediction error of the the i-th obstacle positions $\forall k \geq \tau$.

Proof: The proof is very similar to the one of Lemma 1 ■

Lemma 3: Let $B_{\rho i} : \mathbb{N} \to \mathbb{R}_+$ be a function defined as:

$$B_{\rho i}(\tau) = \sqrt{\xi_{\rho i}(\tau,\varepsilon)}. \tag{15}$$

$B_{\rho i}(\tau)$ gives an upper bound of the prediction error of the the i-th obstacles' radius.

Proof: The proof is very similar to the one of Lemma 1 ■

We use these upper bounds to define a triggering condition in the next section.

Remark 2: Note that $B_q(\tau,\tau) = \sqrt{\xi_q(\tau,\varepsilon)}$ and $B_{oi}(\tau,\tau) = \sqrt{\xi_{oi}(\tau,\varepsilon)}$ and $B_{\rho i}(\tau)$ does not depend on k. For clarity of exposition, we use $B_q(\tau,\tau)$, $B_{oi}(\tau,\tau)$, $B_{\rho i}(\tau)$ in place of $\sqrt{\xi_q(\tau,\varepsilon)}$, $\sqrt{\xi_{oi}(\tau,\varepsilon)}$ and $\sqrt{\xi_{\rho i}(\tau,\varepsilon)}$ respectively throughout the paper.

Remark 3: $\tilde{L}_g$ can be a conservative approximation of the Lipschitz constant of the function $g_i(\cdot)$ to cover the uncertainties of the obstacle dynamics to a satisfying extent.

B. Triggering Conditions

The use of estimated robot and obstacle positions must ensure collision avoidance and destination convergence. At time k, the measurements $y_a(\tau)$, $y_{oi}(\tau)$, and $y_{\rho i}(\tau)$ can be used as long as we guarantee that at time $k+1$ no collision will occur. An example of this scenario is depicted in Fig. 3. If collision avoidance cannot be guaranteed, a new measurement is needed. An illustration of these sporadic measurements is shown in Fig. 4 where the uncertainty bounds $B_q(k,\tau)$ and $B_{oi}(k,\tau)$ increase over time with each step without measurements. The bounds increase until

[1] Throughout the paper we use the notation $\hat{x}(k|k')$ to represent the estimation of $x(k)$ given the measurements at time k'.

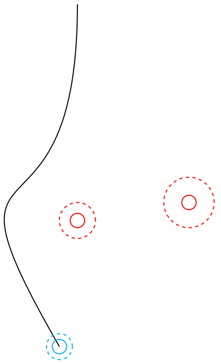
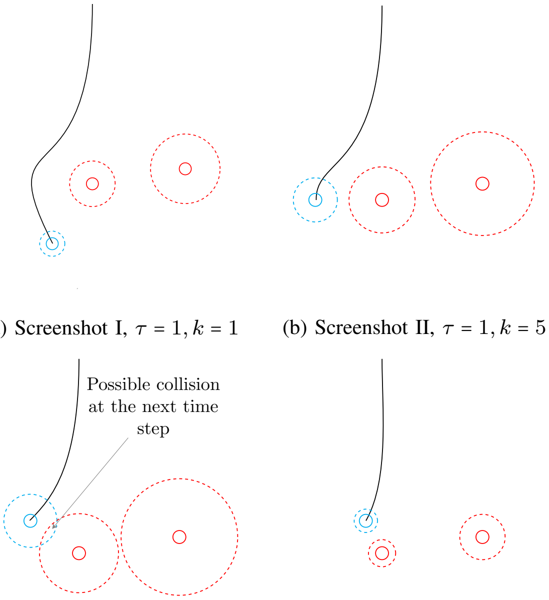


Fig. 3. Example scenario. The blue solid circle is the agent, the red solid circles are the obstacles and the black curve is the collision free path computed. Dashed circles represent the uncertainty in the predicted position at the next time step.



(a) Screenshot I, $\tau = 1, k = 1$

(b) Screenshot II, $\tau = 1, k = 5$

(c) Screenshot III, $\tau = 1, k = 10$

(d) Screenshot IV, $\tau = 10, k = 11$

Fig. 4. Triggers during navigation. The cyan solid circle represents the agent where the red solid circles represent the obstacle. The dashed circles represent the uncertainty in the measure (i.e., those circles are bounding boxes of the predicted obstacle and robot positions at the next time step)

Fig. 4(c) where a new measurement is needed to guarantee collision avoidance. Here we note that there is an overlap of the uncertainties of the next robot position and of the obstacle positions, which means that at the next time step (i.e, $k+1$) there are chances of a collision. Then a new measurement is taken in Fig. 4(d) and the uncertainties are reduced.

The condition for when a new measurement is needed to ensure collision avoidance is:

$$\begin{aligned} &||\hat{q}(k+1|\tau) - \hat{o}_i(k+1|\tau)|| - (r + \hat{\rho}_i(k+1|\tau)) + \\ &- B_q(k+1,\tau) - B_{oi}(k+1,\tau) - B_{\rho i}(\tau) \le 0 \end{aligned} \tag{16}$$

with $i = \{1, 2, \ldots, N\}$.

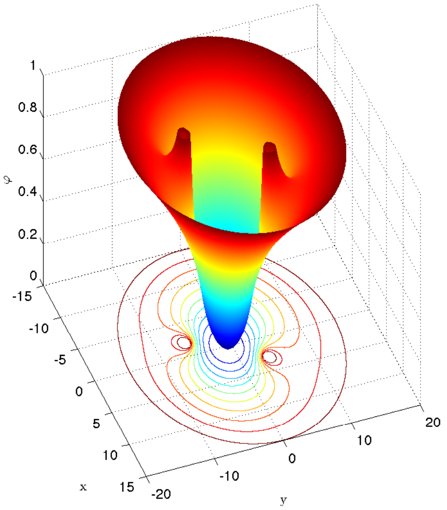


Fig. 5. Example of a navigation function with two spherical obstacles and a spherical workspace.

The condition for when a new measurement is needed to satisfy destination convergence is:

$$||\hat{q}(k|\tau) - d|| - B_q(\tau,\tau) \le 0. \tag{17}$$

We use these triggering conditions to reduce the computational cost of the overall system.

C. Collision Free Navigation

As mentioned earlier, in this paper the collision free navigation is carried out using a NF. This is a continuous function $\varphi : \mathcal{W} \to \Phi$ with $\Phi = [0\ 1]$ reaching a unique and therefore global minimum only at the destination point d and it reaches its maximum on the edges of the obstacles. An example NF in two dimensions is depicted in Fig. 5, but note that the approach holds for arbitrary dimensions and different obstacle shapes [10]. Intuitively, starting from a collision free position, to guarantee destination convergence it is sufficient to decrease the value of the NF at each time step. The chosen NF is defined as follows

$$\varphi(\hat{q}(k|\tau)) \triangleq \left(\frac{\gamma_d(k,\tau)^h}{\gamma_d(k,\tau)^h + \beta(k,\tau)} \right)^{\frac{1}{h}} \tag{18}$$

where the obstacle avoidance is taken into account through an obstacle function $\beta(k,\tau)$ accounting for the uncertainties at time k defined as follows:

$$\begin{aligned} \beta(k,\tau) \triangleq \prod_{i=0}^{N} &||\hat{q}(k|\tau) - \hat{o}_i(\tau|\tau)||^2 - (r + \hat{\rho}_i(\tau|\tau) + \\ &+ B_q(\tau,\tau) + B_{oi}(\tau,\tau) + B_{\rho i}(\tau))^2 \end{aligned} \tag{19}$$

and the destination convergence is taken into account through a destination function γ_d accounting for the uncertainties at time k defined as follows:

$$\gamma_d(k,\tau) \triangleq ||\hat{q}(k|\tau) - d||^2 \tag{20}$$

and h is a shaping parameter that modifies how close to an obstacle the robot moves.

We use φ to ensure convergence of the estimated robot's position.

D. Control Synthesis

Exploiting the properties of NFs, we derive a control input that decreases the value of (18) at each time step. In this case we can ensure that the robot does not hit an obstacle and its final estimated position converges to the desired destination which, under weak assumptions, satisfies the goals of Problem 1. The control input is:

$$u(q(k)) = -B^{\dagger}(f(\hat{q}(k|\tau)) - \bar{q}(k+1|\tau)) \tag{21}$$

where $\bar{q}(k|\tau) = \underset{\|\hat{q}(k|\tau)-\hat{q}'(k|\tau)\|\leq B_q(k+1,\tau)}{\text{argmin}} \bar{\varphi}(\hat{q}'(k|\tau))$,

$$\bar{\varphi} = \underset{\|\hat{q}(k|\tau)-\hat{q}'(k|\tau)\|\leq B_q(k+1,\tau)}{max} (\varphi(\hat{q}'(k|\tau)))$$

and $B^{\dagger}$ is the Moore-Penrose matrix pseudo-inverse of B. Under controllability assumptions, this pseudo-inverse exits.

Remark 4: Note that (21) has only known elements.

Remark 5: The value of $\bar{u}(k)$ used in (9) is:

$$\bar{u}(k) = (1 + L_f)\sqrt{\xi_q(k,\varepsilon)}$$

V. Convergence Analysis

In this section, we show that the controller (21) provides a solution to Problem 1. The main result is presented in Proposition 1 and proved using Lemmas 4-6. Before introducing the lemmas, we state the following assumptions on the workspace:

Assumption 1: The values of ε, $\bar{v}$, $\bar{\omega}_i$, and $\bar{\psi}_i$ are such that $B_q(\tau,\tau), B_{oi}(\tau,\tau), B_{\rho i}(\tau)$ are small enough so that for each pair of obstacles i, j the following condition holds:

$$\begin{aligned}&\|o_i(\tau) - o_j(\tau)\| > \\ &>(\rho_i + \rho_j + B_{oi}(\tau,\tau) + B_{\rho i}(\tau) + B_{oj}(\tau,\tau) + B_{\rho j}(\tau,\tau)+ \\ &\quad + B_q(\tau+1,\tau)).\end{aligned}$$

Under this assumption the NF (18) is free of local minima [4].

Assumption 2: The values of ε, $\bar{v}$, $\bar{\omega}_i$, and $\bar{\psi}_i$ are such that $B_q(\tau,\tau) \leq B_d$. That is, the estimated final position can be used to satisfy the goals of Problem 1.

Assumption 3: The values of ε, $\bar{v}$, $\bar{\omega}_i$, $\bar{\psi}_i$ and the initial robot position is such that $\|\hat{q}(0|0) - \hat{o}_i(0|0)\| \geq (r + \rho_j + B_{oi}(0,0)+B_{\rho i}(0)+B_q(0,0))$ $\forall i = \{0,1,\ldots,N\}$. That is the estimated position can ensure that the robot is in a collision free situation at its initial position.

Assumption 4: The values of ε, $\bar{v}$, $\bar{\omega}_i$, $\bar{\psi}_i$ are such that $B_q(\tau,\tau), B_{oi}(\tau,\tau), B_{\rho i}(\tau)$ are small enough and the destination point d is such that

$$\begin{aligned}&\exists\, T \in \mathbb{N} : \|o_i(t) - d\| \geq r + \rho_j + B_{oi}(t,\tau) + B_{\rho i}(\tau)+ \\ &\quad + B_q(t,\tau)\ \forall t \geq T\ \forall i = \{0,1,\ldots,N\}\end{aligned}$$

that is, from T onward, the region that an obstacle could potentially occupy never intersects the destination d. The convergence is then possible.

Lemma 4: The triggering condition (17) at time k ensures the collision avoidance at time $k+1$.

Proof: One of the following situations occurs:

The triggering condition is satisfied: Then a new measurement takes place, setting $\tau = k$. By Assumptions 1 and 3, using the control input (21) the collision avoidance is guaranteed.

The triggering condition is not satisfied: Then the collision avoidance is guaranteed for the next time step $k+1$ (i.e., $\|q(k+1)-o_i(k+1)\| > (r+\rho_i(k))$). if the triggering condition not is satisfied, then

$$\begin{aligned}&\|\hat{q}(k+1|\tau) - \hat{o}_i(k+1|\tau)\| - (r + \hat{\rho}_i(k+1|\tau))+ \\ &- B_q(k+1,\tau) - B_{oi}(k+1,\tau) - B_{\rho i}(\tau) > 0.\end{aligned} \tag{22}$$

Now adding and subtracting $q(k+1) + o_i(k+1)$ on the first member and $\rho_i(k+1)$ on the second of the left hand side of (22) we get

$$\begin{aligned}&\|\hat{q}(k+1|\tau) - q(k+1) - \hat{o}_i(k+1|\tau) + o_i(k+1) + q(k+1)+ \\ &- o_i(k+1)\| - (r + \rho_i(k+1) - \rho_i(k+1) + \hat{\rho}_i(k+1|\tau))+ \\ &- B_q(k+1,\tau) - B_{oi}(k+1,\tau) - B_{\rho i}(\tau) > 0.\end{aligned}$$

using the triangle inequality:

$$\begin{aligned}&\|\hat{q}(k+1|\tau) - q(k+1)\| + \|-\hat{o}_i(k+1|\tau) + o_i(k+1)\|+ \\ &+\|q(k+1) - o_i(k+1)\| - (r + \rho_i(k+1)) + (-\rho_i(k+1)+ \\ &+\hat{\rho}_i(k+1|\tau)) - B_q(k+1,\tau) - B_{oi}(k+1,\tau)+ \\ &-B_{\rho i}(\tau) > \|\hat{q}(k+1|\tau) - q(k+1) - \hat{o}_i(k+1|\tau)+ \\ &+o_i(k+1) + q(k+1) - o_i(k+1)\| - (r + \rho_i(k+1) \\ &-\rho_i(k+1) + \hat{\rho}_i(k+1|\tau))+ \\ &-B_q(k+1,\tau) - B_{oi}(k+1,\tau) - B_{\rho i}(\tau) > 0.\end{aligned}$$

from (9)-(15)

$$\begin{aligned}&\|q(k+1) - o_i(k+1)\| - (r + \rho_i(k+1)) - 2B_{\rho i}(\tau) > \\ &>\|\hat{q}(k+1|\tau) - q(k+1)\| + \|-\hat{o}_i(k+1|\tau) + o_i(k+1)\|+ \\ &+\|q(k+1) - o_i(k+1)\| - (r + \rho_i(k+1)) + (-\rho_i(k+1)+ \\ &+\hat{\rho}_i(k+1|\tau)) - B_q(k+1,\tau) - B_{oi}(k+1,\tau)+ \\ &-B_{\rho i}(\tau) > 0.\end{aligned}$$

which yields

$$\|q(k+1) - o_i(k+1)\| > (r + \rho_i(k+1)) \tag{23}$$

■

Lemma 5: If the triggering condition (17) is satisfied then destination convergence is maintained.

Proof: When $\|\hat{q}(k|\tau)-d\|-B_q(\tau,\tau) \leq 0$ holds, then a trigger takes place. After the trigger, $k = \tau$ and one of the following situations occurs:

The estimated robot's position has reached the destination: $\|\hat{q}(k|\tau) - d\| = 0$ then $\hat{q}(k|\tau) = d$. Since $B_q(\tau,\tau) \leq B_d$ by Assumption 2, $\|q(k)-d\| = \|q(k)-\hat{q}(k|\tau)\| \leq B_q(\tau,\tau) \leq B_d$.

The estimated robot's position has not reached the destination and $\|\bar{q}(k|\tau) - \hat{q}(k|\tau)\| = B_q(k+1,\tau)$: Since $B_q(k+1,\tau) > B_q(\tau,\tau)$ the control input is such that $\|u(q(k))\| = B_q(k+1,\tau)$ and it will counteract the effects of disturbance and noises, accounted in $B_q(\tau,\tau)$.

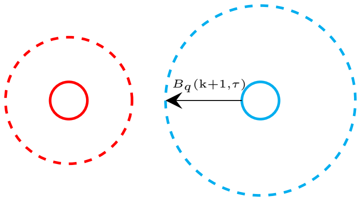


Fig. 6. Illustration of Lemma 6. Solid circles represent the robot (right) and an obstacle (left), dashed circles represent the respective bounding boxes of the predicted next positions.

The estimated robot's position has not reached the destination and $||\bar{q}(k|\tau) - \hat{q}(k|\tau)|| < B_q(k+1,\tau)$: Then the direction that decreases $\bar{\varphi}$ is not on the direction of d hence the robot is close to an obstacle and, by Assumption 2 this cannot be in a B_d-neighborhood of d. ∎

Lemma 6: $\bar{q}$ is always collision free.

Proof: Applying the control input (21) at time k, $\hat{q}(k+1|\tau) = \bar{q}(k|\tau)$ and one of the following cases occurs: The triggering conditions are satisfied: Then a new trigger takes place and $\tau = k$, so $B_q(k,\tau) = B_q(\tau,\tau)$ then the new control input computed according to the new measurements, i.e.

$$\bar{q}(k|\tau) = \underset{||\hat{q}(k|\tau) - \hat{q}'(k|\tau)|| \leq B_q(\tau+1,\tau)}{\operatorname{argmin}} \bar{\varphi}(\hat{q}'(k|\tau))$$

and by Assumptions 1 and 2 a solution exists.
The triggering conditions are not satisfied: Then the collision avoidance is guaranteed by Lemma 4. ∎
Lemma 6 is illustrated in Fig. 6. The triggering condition (17) ensures that the areas where robot and any obstacle can be do not overlap, hence each point within $B_q(k+1,\tau)+r$ from the robot's estimated position, included $\bar{q}$, is collision free.

Proposition 1: If the robot is controlled by (21), then for initial condition satisfying Assumption 3, Problem 1 is solved.

Proof: Let $V(k) = \bar{\varphi}(\hat{q}(k|\tau))$ be a candidate Lyapunov function. If a trigger takes place, $B_q(\tau,\tau) \leq B_q(k+1,\tau)$, then $\bar{\varphi}(\hat{q}(k+1|\tau)) \leq \bar{\varphi}(\hat{q}(\tau|\tau))$. If a trigger does not take place then the controller (21) ensures that $\bar{q}(k|\tau) = \hat{q}(k+1|\tau)$, thus $\bar{\varphi}(\hat{q}(k+1|\tau)) = \bar{\varphi}(\bar{q}(k|\tau))$. In both cases the following holds:

$$\bar{\varphi}(\hat{q}(k+1|\tau)) = \bar{\varphi}(\bar{q}(k|\tau)) \leq \bar{\varphi}(\hat{q}(k|\tau)) \tag{24}$$

and $\hat{q}(k|\tau)$ is always collision free due to Lemma 6. The inequality of (24) ensures that $\bar{\varphi}(\hat{q}(k+1|\tau))$ is not increasing. Then, under Assumption 3, $V(k+1) \leq V(k)$ and $V(k+1) = V(k) \iff ||\hat{q}(k) - d|| = 0$.

From (24) and due to the properties of (18), the set Φ is a positively invariant set of $V(k)$ which is also closed and bounded. By LaSalle's invariance principle [34], the system converges to the largest invariant subset of Φ. The control design implies that $V(k+1) = V(k) \iff ||\hat{q}(k) - d|| = 0$ since $\bar{\varphi}$ cannot decrease any further. Thus the destination point d is the largest invariant set of Φ and $\hat{q}$ converges to d. The triggering conditions do not affect destination convergence and collision avoidance due to Lemmas 4 and 5. Under Assumption 2, when $||\hat{q}(k|\tau) - d|| = 0$ then $||q(k) - d|| \leq B_d$ and the robot is inside the desired margin. ∎

VI. Simulations

To illustrate the convergence and stability of our approach, alongside the proofs presented in Section IV, two numerical solution in a static environment and one in a dynamic environment are shown. The simulations are carried out using MATLAB scripts with an optimizer algorithm to compute the *argmin* function. The algorithm chosen is the *active-set* due to its high computational performance in box constrained problems. For each simulation, a comparison of our approach is made with a traditional one which uses the same NF whereas the feedback measurements are made in a periodic manner. Table I contains some key parameters for each simulation. Figs. 7-9 show the paths executed in each simulation using our approach (solid black line) and the traditional approach (dashed gray line), where the obstacles are depicted as red spheres in their final position, note that they are drawn according to estimated positions to resemble a real scenario. The magenta squares represent positions where a trigger took place and the destination margin B_d is depicted as a semi-transparent green sphere. For completeness, Figs. 10-12 shows the projections of the paths executed on the x-z and x-y axis. Fig. 13 compares the number of feedback measurements executed by our approach and the traditional approach for each simulation. A comparison in dynamic environments is not possible since the traditional approach does not make use of the prediction proposed in Section IV.

In Simulation I the obstacles are far apart which results in a few measurements needed to ensure a collision free navigation. Simulation II presents more measurements than the previous case, since the robot navigates closer to the obstacles due to a higher parameter h used in (18) and described at the end of Section IV-C. Simulation III presents an example of a dynamic environment. One of the obstacles is moving and its trajectory is depicted as a straight red line in Fig. 9. We note how new measurements take place when the obstacle is moving near the robot and its trajectory changes accordingly. Then, further measurements are no longer needed to avoid the collision with the moving obstacle since it drifts away from the robot. We note that the control system does not require a dynamic model of the obstacle, but only a conservative approximation of its Lipschitz constant.

A video showing some other simulations can be found online[2].

Parameter	Simulation I	Simulation II	Simulation III
h	20	80	80
L_f	1.2	1.2	1.2
$\tilde{L}_g$	1	1	1.8
$\xi_q(\tau,\varepsilon)$	$2 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	$2 \cdot 10^{-3}$
$\xi_{oi}(\tau,\varepsilon)$	10^{-2}	10^{-2}	10^{-2}
$\xi_{\rho i}(\tau,\varepsilon)$	10^{-3}	10^{-3}	10^{-3}

TABLE I. Table collecting keys parameters of the simulations.

[2] YouTube video title: Robot Navigation Under Uncertainties.
YouTube video link: http://youtu.be/XA8igLnohSY .

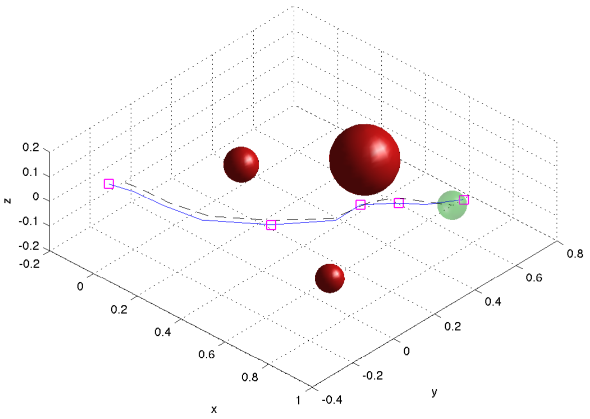


Fig. 7. Path executed in Simulation I. The solid line represents the estimated path, the dashed line represents the real path. The dark red spheres represent the obstacles, the bright sphere represent the destination area. Magenta squares are the position where a sample takes place

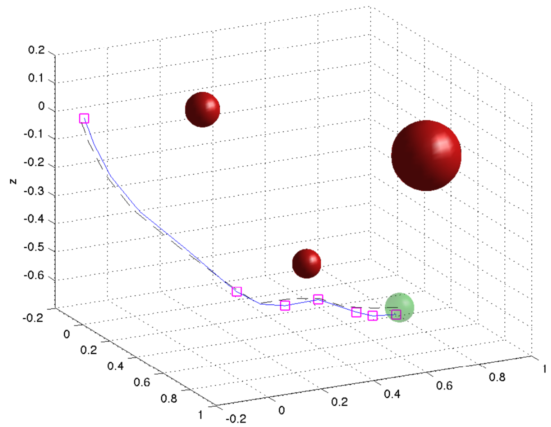


Fig. 8. Path executed in Simulation II. The solid line represents the estimated path, the dashed line represents the real path. The dark red spheres represent the obstacles, the bright sphere represent the destination area. Magenta squares are the position where a sample takes place

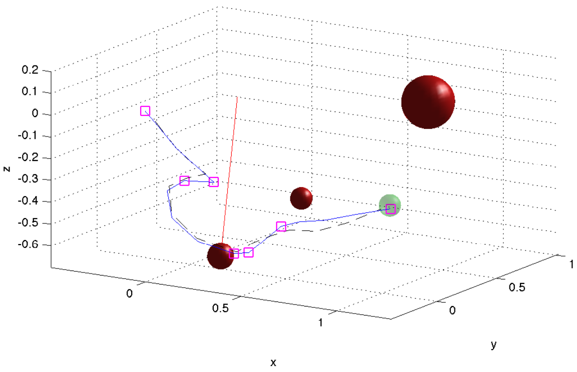
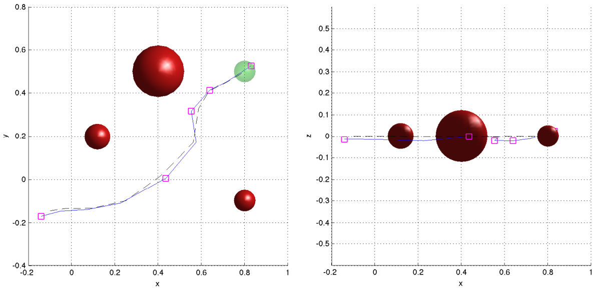


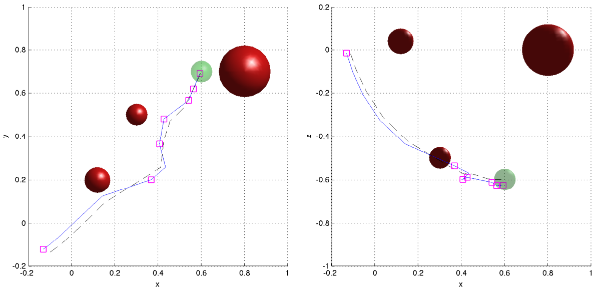
Fig. 9. Path executed in Simulation III. The solid line represents the estimated path, the dashed line represents the real path, the straight line represents an obstacle's path. The dark red spheres represent the obstacles, the bright sphere represent the destination area. Magenta squares are the position where a sample takes place.



(a) Projection on the x-y axis.

(b) Projection on the x-z axis.

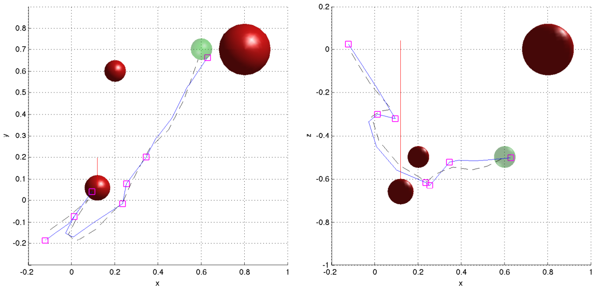
Fig. 10. Projections of the path executed in Simulation I.



(a) Projection on the x-y axis.

(b) Projection on the x-z axis.

Fig. 11. Projections of the path executed in Simulation II.



(a) Projection on the x-y axis.

(b) Projection on the x-z axis.

Fig. 12. Projections of the path executed in Simulation III.

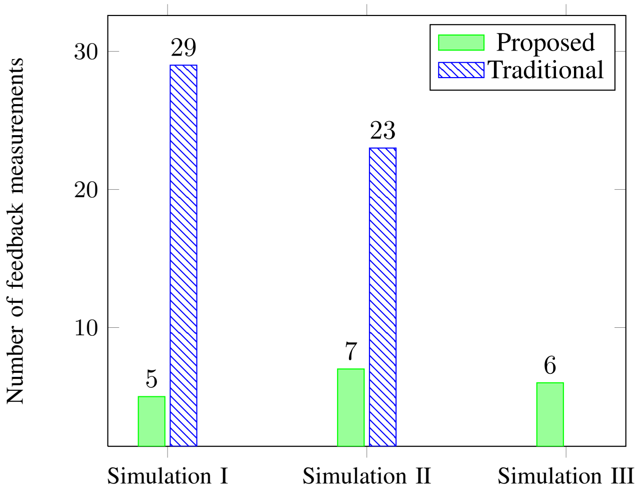


Fig. 13. Comparison of the proposed approach with a traditional one. It is clear how the number of feedback measurements is reduced.

VII. Conclusions

Combining a potential field approach with event driven control, we have shown how to reduce the computational cost of the overall control system, proving properties in terms of goal convergence and collision avoidance. Due to the fact that the calculation of the triggering condition is less demanding than the acquisition, filtering and computation of the sensors' data, the proposed approach is suitable for, among others, real-time systems; high speed vehicles; and tele-operated robots.

Acknowledgment

This work has been supported by the Swedish Research Council (VR) and the European Union Project RECONFIG, (FP7-ICT-2011-9), the authors gratefully acknowledge the support. We also thank Alejandro Marzinotto for his useful and critical review.

References

[1] P. Ogren and N. E. Leonard, "Obstacle avoidance in formation," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 2492–2497.

[2] G. Roussos, D. V. Dimarogonas, and K. J. Kyriakopoulos, "3D navigation and collision avoidance for nonholonomic aircraft-like vehicles," *Int. J. Adapt. Control Signal Process.*, p. n/a, 2010.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[4] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Adv. Appl. Math.*, vol. 11, no. 4, pp. 412–442, Dec. 1990. [Online]. Available: http://dx.doi.org/10.1016/0196-8858(90)90017-S

[5] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 1, pp. 53–64, 2003.

[6] M. Colledanchise, D. Dimarogonas, and P. Ogren, "Obstacle Avoidance in Formation Using Navigation-like Functions and Constraint Based Programming," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 5234–5239.

[7] M. C. De Gennaro and A. Jadbabaie, "Formation control for a cooperative multi-agent system using decentralized navigation functions," in *American Control Conference, 2006*. IEEE, 2006, pp. 6–pp.

[8] P. Ogren, "Split and join of vehicle formations doing obstacle avoidance," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 2. IEEE, 2004, pp. 1951–1955.

[9] P. Ogren and N. E. Leonard, "A convergent dynamic window approach to obstacle avoidance," *Robotics, IEEE Transactions on*, vol. 21, no. 2, pp. 188–195, 2005.

[10] I. F. Filippidis and K. J. Kyriakopoulos, "Navigation functions for everywhere partially sufficiently curved worlds," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2115–2120.

[11] M. Guo, K. Johansson, and D. Dimarogonas, "Motion and action planning under ltl specifications using navigation functions and action description language," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 240–245.

[12] D. V. Dimarogonas, E. Frazzoli, and K. Johansson, "Distributed event-triggered control for multi-agent systems," *Automatic Control, IEEE Transactions on*, vol. 57, no. 5, pp. 1291–1297, May 2012.

[13] X. Wang and M. Lemmon, "Event design in event-triggered feedback control systems," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, Dec 2008, pp. 2105–2110.

[14] K. J. Astrom, "Event Based Control," in *Analysis and Design of Nonlinear Control Systems*, A. Astolfi and L. Marconi, Eds. Springer Berlin Heidelberg, 2008, pp. 127–147.

[15] A. Anta and P. Tabuada, "To Sample or not to Sample: Self-Triggered Control for Nonlinear Systems," *Automatic Control, IEEE Transactions on*, vol. 55, no. 9, pp. 2030–2042, Sept 2010.

[16] S. Durand and N. Marchand, "Further Results on Event-Based PID Controller," in *Proceedings of the European Control Conference 2009*, Budapest, Hongrie, Aug. 2009, pp. 1979–1984, département Automatique Département Automatique.

[17] K.-E. Arzen, "A simple event-based pid controller," in *Proc. 14th World Congress of IFAC, Beijing*, vol. Q, March 1999, pp. 423–428.

[18] L. Bao, M. Skoglund, and K. Johansson, "Encoder decoder design for event-triggered feedback control over bandlimited channels," in *American Control Conference, 2006*, June 2006, pp. 4183–4188.

[19] P. Wan and M. Lemmon, "Optimal power flow in microgrids using event-triggered optimization," in *American Control Conference (ACC), 2010*, June 2010, pp. 2521–2526.

[20] L. Liu, J. Li, Z. Wu, and J. Shu, "Research on warning mechanism of link quality for event-driven wireless sensor network," in *Electronic Commerce and Security, 2009. ISECS '09. Second International Symposium on*, vol. 2, May 2009, pp. 510–515.

[21] M. Mazo and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, Dec 2008, pp. 435–440.

[22] M. Lemmon, "Event-triggered feedback in control, estimation, and optimization," in *Networked Control Systems*, ser. Lecture Notes in Control and Information Sciences, A. Bemporad, M. Heemels, and M. Johansson, Eds. Springer London, 2010, vol. 406, pp. 293–358.

[23] R. Anderson, D. Milutinovic, and D. V. Dimarogonas, "Self-triggered stabilization of continuous stochastic state-feedback controlled systems," in *Control Conference (ECC), 2013 European*, July 2013, pp. 1151–1155.

[24] S. Maniatopoulos, D. Dimarogonas, and K. Kyriakopoulos, "A decentralized event-based predictive navigation scheme for air-traffic control," in *American Control Conference (ACC), 2012*, June 2012, pp. 2503–2508.

[25] D. V. Dimarogonas, "L2 gain stability analysis of event-triggered agreement protocols," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, Dec 2011, pp. 2130–2135.

[26] M. C. F. Donkers and W. P. M. H. Heemels, "Output-based event-triggered control with guaranteed l-gain and improved event-triggering," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, Dec 2010, pp. 3246–3251.

[27] J. Barradas Berglind, T. Gommans, and W. Heemels, "Self-triggered MPC for Constrained Linear Systems and Quadratic Costs," in *IFAC Conference on Nonlinear Model Predictive Control 2012, Noordwijkerhout, Netherlands*, 2012, pp. 342–348.

[28] A. Albert, "Comparison of event-triggered and time-triggered concepts with regard to distributed control systems."

[29] K. Astrom, Karl J and B. Bernhardsson, "Comparison of riemann and lebesgue sampling for first order stochastic systems," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 2, 2002, pp. 2011–2016 vol.2.

[30] J. H. Sandee, P. M. Visser, and W. P. M. H. Heemels, "Analysis and experimental validation of processor load for event-driven controllers," in *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, 2006, pp. 1879–1884.

[31] J. Sandee, W. Heemels, and P. Bosch, *Case Studies in Event-Driven Control*, ser. Lecture Notes in Computer Science, A. Bemporad, A. Bicchi, and G. Buttazzo, Eds. Springer Berlin Heidelberg, 2007, vol. 4416.

[32] A. Alessandri, M. Baglietto, G. Battistelli, and V. Zavala, "Advances in moving horizon estimation for nonlinear systems," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, Dec 2010, pp. 5681–5688.

[33] C. V. Rao, J. B. Rawlings, and J. H. Lee, "Constrained linear state estimationa moving horizon approach," *Automatica*, vol. 37, no. 10, pp. 1619–1628, Oct. 2001.

[34] V. Sundarapandian, "An invariance principle for discrete-time nonlinear systems," *Applied Mathematics Letters*, vol. 16, no. 1, pp. 85 – 91, 2003.