

Design of nonlinear MPC by Kriging-based optimization

Julien Marzat, H el ene Piet-Lahanier

ONERA – The French Aerospace Lab, F-91123 Palaiseau, France,
julien.marzat@onera.fr, helene.piet-lahanier@onera.fr

Abstract: This paper investigates the on-line design of nonlinear model predictive control by the use of Kriging as a surrogate model for optimization. This representation is able to address various optimization problems at a very reduced computational cost. It is thus advocated here to deal with nonlinear models in the context of NMPC. One of the main objectives is to assess whether this strategy may be processed in real-time, while ensuring accurate control. An application to the guidance law design of an unmanned aerial vehicle is reported to illustrate the resulting performance.

Keywords: efficient global optimization, Gaussian process, optimal guidance, Kriging, nonlinear model predictive control, unmanned aerial vehicle.

1. INTRODUCTION

Model Predictive Control (MPC) uses a dynamical model of a system to predict its future state on a time horizon. Using this prediction, an open-loop performance criterion is optimized at each time step to determine the sequence of control inputs that should be applied to fulfill a desired objective (Findeisen et al. (2003)). This scheme makes it possible to take into account an accurate model of the evolution of the system as well as changes in the environment, since new control inputs are computed on the basis of measurements acquired in real time. Many applications have been reported (Garcia et al. (1989)), the majority being based on linear models while a few use nonlinear ones (Qin and Badgwell (2003)).

Nonlinear model predictive control (NMPC) has raised many theoretical and practical problems. First of all, unlike linear models, it is very difficult to obtain an analytical solution to the optimization problem that provides the control inputs. A lot of approximation techniques have been developed, most of them being based on linearization or parameterization of the state or input spaces (see, e.g., Singh and Fuller (2001) or Kang and Hedrick (2009)) or learning techniques (e.g., neural networks in Witt et al. (2007)). This may lead to a loss of accuracy, but also to a heavy computational cost, which is cumbersome regarding real-time processing. A potential way of overcoming these issues is to use a surrogate model for representing the criterion evolution.

Kriging is a statistical interpolation tool using Gaussian Processes (GP) to approximate continuously a function whose values are known at sampled input points. It also provides a level of confidence in this prediction (Kleijnen (2009)). This additional knowledge has made it possible to design an iterative procedure that looks for the global optimum of the function via a trade-off between the exploration of unknown areas and local search. This algorithm, named *Efficient Global Optimization* (EGO)

by Jones et al. (1998), showed the ability to address multivariate optimization at a low computational cost, which can be quantified in terms of number of evaluations of the optimized function (see, e.g., Sasena (2002); Santner et al. (2003)). These tools seem to be promising contenders to solve complex nonlinear optimization problem in reasonable time and with sufficient accuracy.

The use of GP was investigated in Deisenroth et al. (2009) for approximating the functions involved in dynamic programming on state and input space samplings, for optimal control purpose. This was shown to yield convincing results for a LQ problem and a pendulum swing-up. However, the potential of Kriging-based optimization was not exploited to obtain the control commands. Another application, involving EGO this time, has been reported in Defretin et al. (2010) to tackle the active recognition of an object from a database via reinforcement learning. This setup improved the recognition performance compared to state-of-the-art solutions, under strict computational constraints.

In this paper, we propose to profit from the properties of Kriging and EGO to address the on-line implementation of nonlinear model predictive control. The idea is to approximate by a Kriging model the function between the vector of inputs on the prediction horizon and the quality criterion to be optimized. At each time step, the computation of this function implies the numerical integration of the nonlinear dynamical model on this prediction horizon. One of the main interests of EGO is that the budget of evaluations of the function optimized can be fixed by the user, which can be used to maintain the computation time of one step of NMPC at a reasonable value. An approach similar in spirit was introduced in Frew (2005), where a random sampling of the input space was used to pick a suboptimal solution to the control problem. However, there was no iterative sampling policy to improve the estimate of the optimum, unlike the procedure we propose that makes an efficient use of the simulation budget allotted.

The basics of Kriging and EGO are described in Section 2, along with the NMPC algorithm proposed. In Section 3, an application of this new scheme to the guidance of an Unmanned Aerial Vehicle (UAV) is presented. Several tests have been performed to assess the accuracy of the response and the computational load of the method. The guidance of the vehicle with obstacle avoidance capability is also reported. Conclusions and perspectives are discussed in Section 4.

2. METHOD DESCRIPTION

2.1 Nonlinear model predictive control

Model predictive control computes the control input by solving on-line an optimization problem over a finite horizon of time. Only a small part of the optimal control sequence is applied to the system, and the computation is repeated at the following instant in a receding way (Garcia et al. (1989)). Although such approaches require on-line computation, they are very interesting for their ability to directly handle constraints on the input or on the state of the system. A local optimality can be guaranteed with respect to a given performance criterion and the control action also takes into account a prediction of the behavior of the system, which can be useful in the case of trajectory tracking. It can be used to control continuous nonlinear systems by using a discrete representation of the control inputs, which is very well suited for practical implementation. Closed-loop stability can be guaranteed by using a state constraint that imposes norm contractions of the predicted states of the system.

Consider that the system dynamics is described by the state equation

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad \forall t > 0 \\ \mathbf{x}(0) &= \mathbf{x}_0 \end{aligned} \quad (1)$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the system state vector and $\mathbf{u}(t) = [u^1(t), \dots, u^m(t)]^T \in \mathbb{R}^m$ its control input vector. The nonlinear mapping \mathbf{f} governs the dynamics of the system. At any instant t on a time horizon $T > 0$, the prediction model is

$$\begin{aligned} \dot{\bar{\mathbf{x}}}(\tau) &= \bar{\mathbf{f}}(\bar{\mathbf{x}}(\tau), \mathbf{u}(\tau)) \quad \forall \tau \in [t, t+T] \\ \bar{\mathbf{x}}(t) &= \mathbf{x}(t) \end{aligned} \quad (2)$$

where $\bar{\mathbf{x}}$ is the predicted state. The cost function associated to the desired performance of the system is defined as

$$y(t, \bar{\mathbf{x}}(t), T, \mathbf{u}(\cdot)) = \int_t^{t+T} q(\bar{\mathbf{x}}(\tau), \mathbf{u}(\tau)) d\tau + V(\bar{\mathbf{x}}(t+T)) \quad (3)$$

where $q(\cdot, \cdot)$ is a positive semi-definite form defined on $\mathbb{R}^{n_x} \times \mathbb{R}^m \rightarrow \mathbb{R}^+$ and $V(\cdot)$ a positive definite form on $\mathbb{R}^{n_x} \rightarrow \mathbb{R}^+$. At time t , the control is chosen so as to minimize y , under constraints on the state and/or on the input vectors. Let $\mathbf{u}_t^*(\cdot)$ be the control, solution of the optimization problem

$$\begin{aligned} \mathcal{P}(t, \mathbf{x}(t)) : \quad & \min_{\mathbf{u}(\cdot)} y(t, \bar{\mathbf{x}}(t), T, \mathbf{u}(\cdot)) \\ & \text{such that } \bar{\mathbf{x}}(\tau) \in \mathbb{X}, \quad \forall \tau \in [t, t+T] \\ & \mathbf{u}(\tau) \in \mathbb{U}, \quad \forall \tau \in [t, t+T] \end{aligned} \quad (4)$$

where $\mathbb{X} \subset \mathbb{R}^{n_x}$ and $\mathbb{U} \subset \mathbb{R}^m$ are the constraints on the state and the control input vectors. The predicted state $\bar{\mathbf{x}}$

is also constrained by the prediction model (2). According to the receding horizon scheme, only $\mathbf{u}_t^*(\tau)$ for $\tau \in [t, t+\delta]$ is applied to the system, where $0 < \delta \leq T$ is the sampling time step at which the control input is applied to the system. This defines the number of prediction steps $h = T/\delta$ on the time horizon T . The whole procedure is repeated at the next sampled instant of time to compute the following control input vector. Note that the optimization process used to compute the control can be stopped as soon as a solution that satisfies the stabilizing constraint has been found. Hence suboptimal solutions may be considered to reduce the online computation burden.

On the time horizon T , the computation of y from (4) depends on h successive control input vectors \mathbf{u} , which are aggregated in \mathbf{u}_h such that

$$\mathbf{u}_h = \begin{bmatrix} \mathbf{u}(t) \\ \mathbf{u}(t+\delta) \\ \vdots \\ \mathbf{u}(t+T-\delta) \end{bmatrix}$$

$$= [u^1(t), \dots, u^m(t), \dots, u^1(t+T-\delta), \dots, u^m(t+T-\delta)]^T \quad (5)$$

The dimension of this vector is equal to $m \times h$, namely the product of the number of control inputs with the number of prediction steps. The computation of y also requires the numerical integration of the prediction model (2) over $[t, t+T]$. Given initial conditions for the predicted state $\bar{\mathbf{x}}(t)$ (e.g., from available sensor data), the cost function (3) can be rewritten as

$$y(t, \bar{\mathbf{x}}(t), T, \mathbf{u}(\cdot)) = y(\mathbf{u}_h). \quad (6)$$

The NMPC algorithm advocated in this paper aims at finding an optimal sequence of control inputs \mathbf{u}_h^* minimizing $y(\cdot)$, under the constraints (2) and (4). A potential approach to provide such solutions within a restricted computation budget (implied by on-line processing) is the use of surrogate models. Among the choice of possible surrogates, Kriging (Lefebvre et al. (1996); Rasmussen and Williams (2006); Kleijnen (2009)) has demonstrated its potential usefulness to address such an optimization problem at a low computational cost.

2.2 Basics of Kriging

Let $\mathbf{y}_n = [y_1, \dots, y_n]^T$ be the criterion values obtained for n simulations of the dynamic system (2) on the time horizon T , corresponding to n initial samples of successive control inputs on \mathbb{U}^h ,

$$\mathcal{U}_n = [\mathbf{u}_h^{(1)}, \dots, \mathbf{u}_h^{(n)}]^T. \quad (7)$$

The Kriging surrogate approximation uses only these n already available results to obtain a simple continuous prediction model that can be processed quickly. The main idea is to model the function $y(\mathbf{u}_h)$ as a zero-mean Gaussian process $Y(\mathbf{u}_h)$ with covariance function $\text{cov}(\cdot, \cdot)$. Kriging is then the search for the *best unbiased linear predictor* of $Y(\cdot)$. Although the actual covariance $\text{cov}(\cdot, \cdot)$ is usually unknown, it can be expressed as

$$\text{cov}\left(Y\left(\mathbf{u}_h^{(i)}\right), Y\left(\mathbf{u}_h^{(j)}\right)\right) = \sigma_Y^2 R\left(\mathbf{u}_h^{(i)}, \mathbf{u}_h^{(j)}\right) \quad (8)$$

where σ_Y^2 is the process variance, to be estimated on the available data, and $R(\cdot)$ a parameterized correlation function to be chosen. The intuitive idea behind the assumption is that, the closer $\mathbf{u}_h^{(i)}$ is to $\mathbf{u}_h^{(j)}$, the closer $y(\mathbf{u}_h^{(i)})$ should be to $y(\mathbf{u}_h^{(j)})$. The classical choice of correlation function adopted in the present paper is the squared exponential correlation function,

$$R(\mathbf{u}_h^{(i)}, \mathbf{u}_h^{(j)}) = \exp\left(-\frac{1}{2}\|\mathbf{u}_h^{(i)} - \mathbf{u}_h^{(j)}\|^2\right). \quad (9)$$

Many other choices of correlation functions are possible (Santner et al. (2003)). The required characteristics for the correlation function are to be equal to 1 when the distance between two points is null, and to tend to zero when this distance increases.

Following these assumptions, the Kriging prediction at any point $\mathbf{u}_h \in \mathbb{U}^h$ is given by

$$\widehat{Y}(\mathbf{u}_h) = \mathbf{r}^T \mathbf{R}^{-1} \mathbf{y}_n \quad (10)$$

where

$$\mathbf{r} = \left[R(\mathbf{u}_h, \mathbf{u}_h^{(1)}), \dots, R(\mathbf{u}_h, \mathbf{u}_h^{(n)}) \right]^T \quad (11)$$

$$\mathbf{R}_{i,j} = R(\mathbf{u}_h^{(i)}, \mathbf{u}_h^{(j)})$$

This interpolating predictor is linear in \mathbf{y}_n , making it light to compute. Another interesting property of Kriging, crucial regarding global search, is the possibility to compute the *variance of the prediction error*

$$\widehat{\sigma}^2(\mathbf{u}_h) = \sigma_Y^2 (1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r}) \quad (12)$$

This provides a statistical confidence level in the prediction, which can be used in particular to indicate the locations in the input space where the original function value is very uncertain.

2.3 Efficient Global Optimization (EGO)

The idea of Kriging-based optimization is to replace the optimization of the function y , which is very expensive to compute, by the optimization of a much lighter function built on the Kriging model. An iterative procedure called EGO has been designed to this end (see Jones et al. (1998)). Given the past results \mathbf{y}_n obtained at \mathcal{U}_n , it uses the knowledge of the variance of the prediction error (12) and the Kriging prediction (10) to suggest an $(n+1)$ -th point according to a criterion measuring the interest of an additional evaluation to improve the optimum estimation. A possible choice for this criterion is the *Expected Improvement* (EI), whose closed-form expression is

$$\text{EI}(\mathbf{u}_h, \widehat{\sigma}, \widehat{Y}, y_{\min}^n) = \widehat{\sigma}(\mathbf{u}_h) [\Phi(w) + w\phi(w)], \quad (13)$$

where

$$w = \frac{(y_{\min}^n - \widehat{Y}(\mathbf{u}_h))}{\widehat{\sigma}(\mathbf{u}_h)}, \quad (14)$$

$$y_{\min}^n = \min_{i=1..n} y(\mathbf{u}_h^{(i)}),$$

and where Φ is the cumulative distribution function and ϕ the probability density function of the normal distribution $\mathcal{N}(0,1)$. The best available estimator of the minimum of y after the first n evaluations is y_{\min}^n . The EI criterion achieves a trade-off between global search (where the uncertainty measure $\widehat{\sigma}$ is high) and local search around

the best current optimum estimate. It is thus well suited for global optimization.

A preliminary sampling step is required to obtain the n points of the initial design \mathcal{U}_n . It has been achieved via Latin Hypercube Sampling (LHS), but other space-filling designs might be considered (McKay et al. (1979)). A common rule of thumb is to choose n equal to 10 times the dimension of the input space (Jones et al. (1998)). For the NMPC problem under study, it yields $n = 10 \times m \times h$.

The description of EGO is given in Algorithm 1. The iterative loop stops when the maximal number of iterations n_{\max} is reached. This number could be adjusted according to the embedded computational resources. In the following application, it has been set to 10 iterations. Our implementation uses the Adaptive Random Search (ARS) optimization algorithm (Pronzato et al. (1984)) to address the maximization of EI (Step 7 of Algorithm 1). The maximal number of calls of the EI function (13) by this global-purpose algorithm can be fixed, which makes it possible to reduce the maximal computation time required per iteration.

Theoretical results on the convergence of EGO have been established recently by Vazquez and Bect (2010) or Bull (2011). They have proven that, if the covariance function is appropriate to the problem under study, the sampling sequence converges at a high rate to the true global optimum. Correlation (9) corresponds to a smooth prior on the cost function to be optimized, which is a reasonable assumption for the real-world problem examined in Section 3.

- (1) Choose $\mathcal{U}_n = \{\mathbf{u}_h^{(1)}, \dots, \mathbf{u}_h^{(n)}\}$ by LHS in \mathbb{U}^h
- (2) Compute $\mathbf{y}_n = \{y(\mathbf{u}_h^{(1)}), \dots, y(\mathbf{u}_h^{(n)})\}$
- (3) $i \leftarrow 0$
- (4) **while** $i < n_{\max}$
- (5) Fit a Kriging model on $\{\mathcal{U}_n, \mathbf{y}_n\}$ with (10)
- (6) Find $y_{\min}^n = \min_{i=1..n} \{y(\mathbf{u}_h^{(i)})\}$.
- (7) Find $\mathbf{u}_h^{(n+1)} = \arg \max_{\mathbf{u}_h \in \mathbb{U}^h} \text{EI}(\mathbf{u}_h, \widehat{\sigma}, \widehat{Y}, y_{\min}^n)$
- (8) Append $\mathbf{u}_h^{(n+1)}$ to \mathcal{U}_n and $y(\mathbf{u}_h^{(n+1)})$ to \mathbf{y}_n
- (9) Go to Step 5 with $n \leftarrow n+1$ and $i \leftarrow i+1$
- (10) **end while**
- (11) Return $\mathbf{u}_h^* = \arg \min_{i=1..n} \{y(\mathbf{u}_h^{(i)})\}$ as a solution to the NMPC optimization problem (4)

Algorithm 1. One step of NMPC with EGO

This procedure is repeated at each time step to find \mathbf{u}_h^* , whose first component $\mathbf{u}(t)$ is then applied as the control input vector of the dynamical system on the time interval $[t, t + \delta]$. The overall stability of the control strategy is ensured by the MPC scheme itself, which should provide a sequence of trajectories with decreasing cost. Moreover, according to Scokaert et al. (1999), a suboptimal solution at each step is sufficient to ensure stability, as long as it remains feasible.

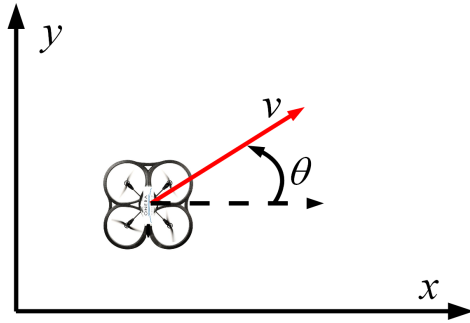


Fig. 1. UAV modeling

3. EXAMPLE OF APPLICATION

3.1 UAV model

We consider the 3-degree-of-freedom lateral motion of an UAV, assuming that its stabilization at a constant altitude is ensured by a faster control loop. The state vector of this dynamical system is $\mathbf{x} = [x, y, \theta]^T$, where $\mathbf{p} = [x, y]^T$ is the position of the vehicle in the horizontal plane, and θ its heading angle (see Figure 1). The control input is $\mathbf{u} = \omega$, while the speed v is assumed to be regulated at the constant value $v_0 = 2.5 \text{ m} \cdot \text{s}^{-1}$. The resulting dynamical model is

$$\begin{cases} \dot{x} = v_0 \cos \theta \\ \dot{y} = v_0 \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (15)$$

This simplified (yet nonlinear) model is often used to illustrate the potential of guidance algorithms (see, e.g., Singh and Fuller (2001); Frew (2005); Kang and Hedrick (2009)). The guidance problem to be solved is the ability of the vehicle to reach a desired position $\mathbf{p}_g = [x_g, y_g]^T$, with minimum control input values. The associated cost function, for h prediction steps, is given by

$$y = \sum_{k=1}^h (\|\mathbf{p}(k) - \mathbf{p}_g\|^2 + \mathbf{u}(k)^2), \quad (16)$$

under the control constraint $\mathbf{u} = \omega \in [-1; 1] \text{ rad} \cdot \text{s}^{-1}$. The initial position of the vehicle in all tests is $\mathbf{p}_i = [2, 3]^T$ and the initial heading angle is $\theta_1 = 1 \text{ rad}$.

3.2 Results and computational load

The tests reported have been performed using a C++ implementation on a 2.2 GHz Intel CPU. In particular, the integration of the nonlinear dynamical model of the system has been achieved with a fourth-order Runge-Kutta scheme.

Figure 2 shows the trajectories obtained with the method proposed to reach $\mathbf{p}_g = [15, 25]^T$. The time horizon is fixed to $T = 1$ second with time steps varying from $\delta = 200 \text{ ms}$ to $\delta = 500 \text{ ms}$, which changes the number of steps h accordingly. It could be seen that an adequate trajectory is obtained for all the combinations of parameters, with also reduced power consumption as control input values remain small (Figure 3).

It should be kept in mind that the NMPC algorithm proposed presents randomness induced by the choice of initial interpolation points at Step 1 of Algorithm 1. To

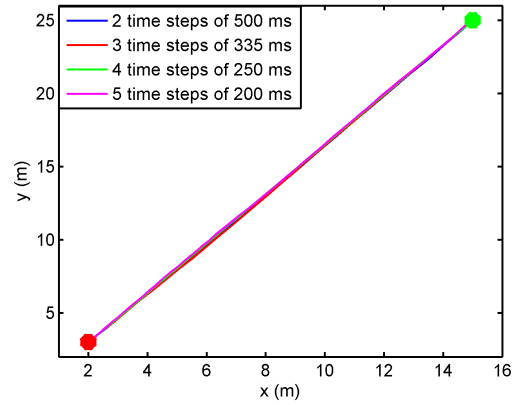


Fig. 2. Optimal guidance with a time horizon of 1 second, for various number and duration of time steps

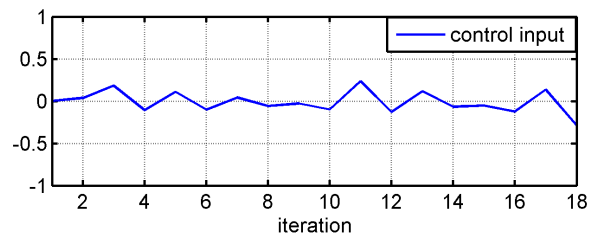


Fig. 3. Evolution of control inputs ($T = 4 \times 250 \text{ ms}$)

illustrate the influence of this randomness, 20 simulations have been run with a prediction horizon of $T = 2 \times 500 \text{ ms}$. The results, displayed on Figure 4, show very small sensitivity to this initialization, since the obtained trajectories are quite close to each other.

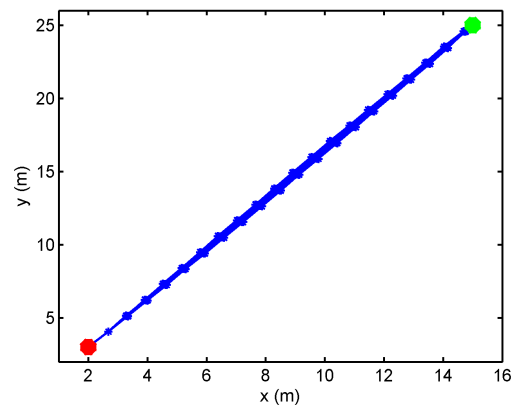


Fig. 4. 20 simulated trajectories ($T = 2 \times 500 \text{ ms}$)

The computation time required for each step of the NMPC algorithm has been evaluated for different parameter choices. The averaged results and the corresponding standard deviations are reported in Table 1. What is most remarkable is that the real-time processing condition is verified for all configurations, within the computation environment mentioned before. This is very interesting, considering that these nonlinear optimization problems are solved on spaces whose dimensions range from 1 to 7. Figure 5 displays the average computation time and associated dispersion for computing one step, as a function of the prediction horizon. The mean computation time and its associated 95% confidence level appear to be significantly

lower than the real-time processing constraint of 500 ms that was allotted in this case. With a larger number of inputs, the same increase in computation time will be observed and thus the number of prediction steps might need to be reduced.

Table 1. Computation time per NMPC step (averaged on 200 iterations)

Optimization parameters	Mean time	Std. dev.
1 time step of 500 ms	22 ms	16 ms
2 time steps of 500 ms	49 ms	28 ms
3 time steps of 335 ms	72 ms	34 ms
3 time steps of 500 ms	97 ms	44 ms
4 time steps of 250 ms	166 ms	115 ms
4 time steps of 500 ms	139 ms	53 ms
5 time steps of 200 ms	92 ms	20 ms
5 time steps of 500 ms	174 ms	39 ms
6 time steps of 500 ms	250 ms	56 ms
7 time steps of 500 ms	393 ms	59 ms

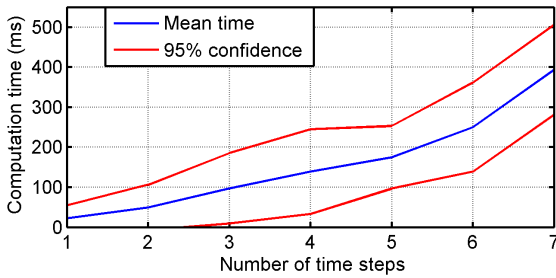


Fig. 5. Computation times per step (time step 500 ms)

3.3 Results with limited control range

We now assume that the control range is severely limited, such that $\mathbf{u} = \omega \in [-0.1; 0.1] \text{ rad} \cdot \text{s}^{-1}$. The optimal guidance algorithm is required to make the vehicle reach the location $\mathbf{p}_g = [30, -20]^T$. On Figure 6, the number of steps is fixed to $h = 2$, while different time steps are tested. The obtained results are consistent with what could have been expected, since the trajectory of the UAV is shorter when the vehicle receives control inputs more frequently. Figure 7 displays the resulting trajectories for several goal locations with the same prediction horizon, $T = 4 \times 250$ ms. The algorithm shows here its ability to cope with the constraints on the control input and to guide successfully the vehicle towards final goals at various locations in space.

3.4 Obstacle avoidance ability

One of the main interest of the NMPC algorithm proposed is that it can deal with non-quadratic or even discontinuous cost functions. Consider for example the same

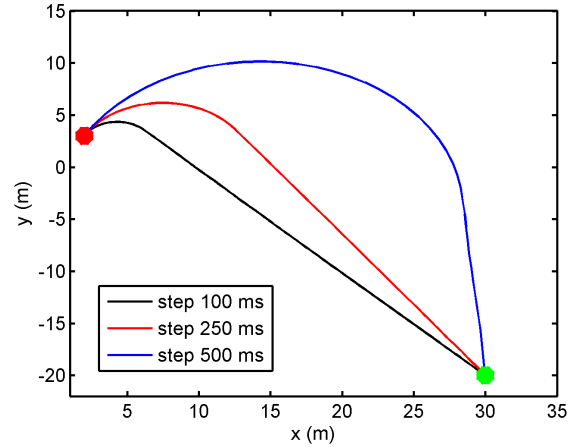


Fig. 6. Limited control range, $h = 2$ and various time steps

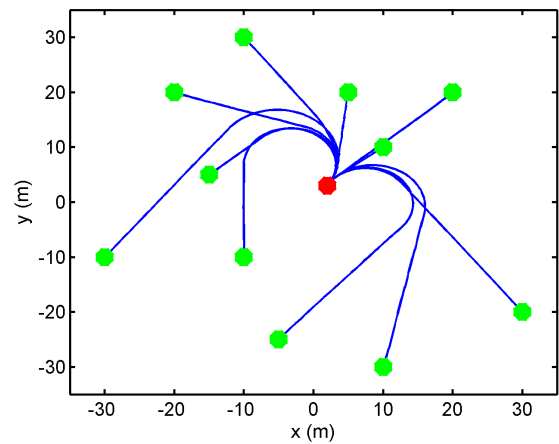


Fig. 7. Limited control range, different goal locations, $T = 2 \times 250$ ms

guidance problem where there exists a forbidden area \mathcal{S} that the vehicle should not cross (green zone in Figure 8). A simple way to deal with this constraint is to penalize the cost function accordingly,

$$y_{\text{obs}} = \sum_{k=1}^h (\|\mathbf{p}(k) - \mathbf{p}_g\|^2 + \mathbf{u}(k)^2 + o(k)) \quad (17)$$

$$\text{where } o(k) = \begin{cases} 1 & \text{if } \mathbf{p}(k) \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases}$$

The membership test of the predicted position of the vehicle with the area \mathcal{S} introduces a discontinuity in the cost function. The resulting trajectory, presented in Figure 8, shows that the NMPC scheme using EGO is indeed able to optimize this non-conventional optimization problem and find an adequate solution for the guidance of the UAV that circumvents the obstacle.

4. CONCLUSIONS AND PERSPECTIVES

This paper has proposed the introduction of Kriging-based optimization within the nonlinear model predictive control framework. The EGO algorithm exploits the statistical properties of the Kriging interpolation to search for a global minimizer of a black-box function under a very restricted budget of evaluations. It thus seems to be a promising way to determine optimal sequences of control

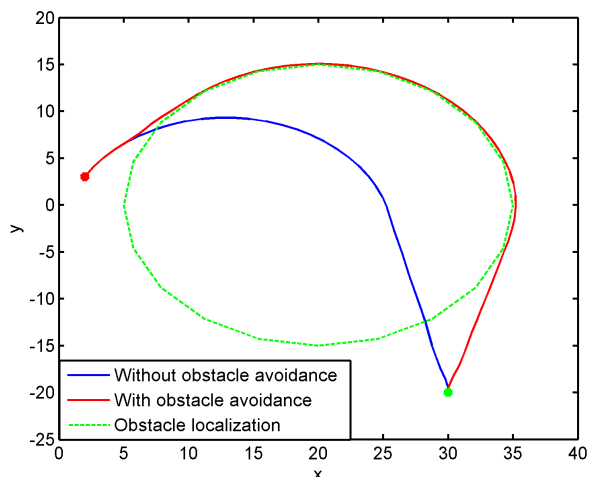


Fig. 8. Illustration of obstacle avoidance ability

inputs minimizing a desired cost function in the context of NMPC. In particular, this does not require the linearization of the dynamical model of the system.

The application of this strategy to the guidance of a reduced-order UAV model has highlighted the potential of this method. Accurate and reproducible results have been obtained, for various configurations of prediction horizon and under strict control constraints. Above all, the computation time required by one iteration of the NMPC procedure is compatible with real-time processing constraints, even if this remains to be assessed by future real embedded tests.

The associated dimension of the input space for the optimization problem varied from 1 to 7 here. This dimension corresponded to the number of time steps considered in the cost function, since the control was unidimensional. The algorithm proposed should be able to deal with higher-dimensional problems. To remain implementable, this may require increasing the control-application time step while keeping it within the range of classical guidance frequencies.

An interesting feature of the method is its ability to deal with non-quadratic and possibly discontinuous cost functions. This has been illustrated through the optimization of a cost function penalizing the crossing of an obstacle area, with successful results. This paves the way for the efficient handling of more complex cost functions for NMPC applications.

REFERENCES

Bull, A.D. (2011). Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12, 2879–2904.

Defretin, J., Marzat, J., and Piet-Lahanier, H. (2010). Learning viewpoint planning in active recognition on a small sampling budget: a Kriging approach. In *Proceedings of the 9th IEEE International Conference on Machine Learning and Applications, ICMLA 2010, Washington DC, USA*, 169–174.

Deisenroth, M.P., Rasmussen, C.E., and Peters, J. (2009). Gaussian process dynamic programming. *Neurocomputing*, 72(7-9), 1508–1524.

Findeisen, R., Imsland, L., Allgower, F., and Foss, B.A. (2003). State and output feedback nonlinear model predictive control: An overview. *European journal of control*, 9(2-3), 190–206.

Frew, E.W. (2005). Receding horizon control using random search for uav navigation with passive, non-cooperative sensing. In *Proceedings of the AIAA Guidance Navigation and Control Conference, San Francisco, USA*, 1–13.

Garcia, C.E., Prett, D.M., and Morari, M. (1989). Model predictive control: Theory and practice—a survey. *Automatica*, 25(3), 335–348.

Jones, D.R., Schonlau, M.J., and Welch, W.J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4), 455–492.

Kang, Y. and Hedrick, J.K. (2009). Linear tracking for a fixed-wing uav using nonlinear model predictive control. *IEEE Transactions on Control Systems Technology*, 17(5), 1202–1210.

Kleijnen, J.P.C. (2009). Kriging metamodeling in simulation: a review. *European Journal of Operational Research*, 192(3), 707–716.

Lefebvre, J., Roussel, H., Walter, E., Lecointe, D., and Tabbara, W. (1996). Prediction from wrong models: the Kriging approach. *IEEE Antennas and Propagation Magazine*, 38(4), 35–45.

McKay, M.D., Beckman, R.J., and Conover, W.J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2), 239–245.

Pronzato, L., Walter, E., Venot, A., and Lebruchec, J.F. (1984). A general-purpose global optimizer: Implementation and applications. *Mathematics and Computers in Simulation*, 26(5), 412–422.

Qin, S.J. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control engineering practice*, 11(7), 733–764.

Rasmussen, C.E. and Williams, C.K.I. (2006). *Gaussian Processes for Machine Learning*. Springer-Verlag, New York.

Santner, T.J., Williams, B.J., and Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag, Berlin-Heidelberg.

Sasena, M.J. (2002). *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD Thesis, University of Michigan, USA.

Scokaert, P.O.M., Mayne, D.Q., and Rawlings, J.B. (1999). Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3), 648–654.

Singh, L. and Fuller, J. (2001). Trajectory generation for a UAV in urban terrain, using nonlinear MPC. In *Proceedings of the IEEE American Control Conference, Arlington, USA*, volume 3, 2301–2308.

Vazquez, E. and Bect, J. (2010). Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference*, 140(11), 3088–3095.

Witt, J., Boonto, S., and Werner, H. (2007). Approximate model predictive control of a 3-DOF helicopter. In *Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, USA*, 4501–4506.