Tracking Model Predictive Control

Daniel Limon and Teodoro Alamo
Departamento de Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla, Spain

Abstract

The main objective of tracking model predictive control is to stabilize the plant satisfying the constraints and steering the tracking error, that is, the difference between the reference and the output, to zero. In order to predict the expected evolution of the tracking error, some assumptions on the future values of the reference must be considered. Since the reference may differ from expected, the tracking problem is inherently uncertain.

The most common case is to assume that the reference will remain constant along the prediction horizon. These predictive control schemes are typically based on a two-layer control structure in which, provided the value of the reference, first an appropriate target equilibrium point is computed, and then an MPC is designed to regulate the system to this target. Under certain assumptions, closed-loop stability can be guaranteed if the initial state is inside the feasibility region of the MPC. However, if the value of the reference is changed, then there is no guarantee that feasibility and stability properties of the resulting control law hold. Specialized predictive controllers have been designed to deal with this problem. Particularly interesting is the so-called MPC for tracking, which ensures recursive feasibility and asymptotic stability of the setpoint when the value of the reference is changed.

The presence of exogenous disturbances or model mismatches may lead to the controlled system to exhibit offset error. Offset-free control can be achieved by using disturbance models and disturbance estimators together with the tracking predictive controller.

Keywords

Reference tracking · Loss of recursive feasibility · MPC for tracking · Offset-free control

Introduction

The problem of designing stabilizing model predictive controllers to regulate a system to the origin has been widely studied, and there are well-known solutions for varied cases including linear, nonlinear, and uncertain systems among others (Rawlings et al. 2017).

The objective of tracking MPC is to ensure that the tracking error, which is the difference between a reference or desired output r and the actual output y, tends to zero.

The most common tracking problem is when the reference r is constant. In this case, the controller is required to steer the state of the plant x and the control input applied to the plant u to the equilibrium point (x_r, u_r) where the tracking error $y - r$ is zero. It is also necessary to ensure that x_r is asymptotically stable for the controlled system, i.e., that the state x converges to x_r and that, near x_r, small changes in x cause small changes in the subsequent trajectory. A relatively straightforward solution for this problem exists (Pannocchia 2004).

Setpoint or constant reference tracking is a relevant control problem in the process industry in which the plant is typically designed to operate at an equilibrium point that maximizes the profit of the plant. In this case, the optimal reference is calculated online by a real-time optimizer (RTO) according to an economic criteria. The reference remains constant for a long period of time, until the RTO, which is executed at a very low frequency, calculates a different one. The equilibrium point associated to the given reference must be calculated and provided to the MPC to steer the plant to this target (Muske 1997).

The tracking problem is considerably more difficult when the reference r varies in a way not known a-priori because MPC is naturally suited to deterministic control problems. As such, in order to deal with varying references, specialized techniques must be developed, a variety of which have been proposed in Pannocchia and Rawlings (2003), Maeder and Morari (2010), Bemporad et al. (1997), Chisci and Zappa (2003), Limon et al. (2008), or Rossiter et al. (1996).

Another tracking problem arises when there exists a mismatch between the model used for prediction in the optimal control problem and the real plant. If the reference is constant and the model mismatch is sufficiently small not to cause loss of asymptotic stability, the state and control will converge to values at which the predicted tracking error, but not the actual tracking error, is zero. The difference between the predicted and actual values of the output y is known as the offset; offset free tracking when the reference is constant may be achieved by the incorporation of a suitable observer to estimate the offset (Pannocchia and Rawlings 2003).

Notation

The set $\mathbb{I}_M$ denotes the set of integers $\{0, 1, \cdots, M\}$. I_n denotes the identity matrix in $\mathbb{R}^{n \times n}$. $\mathbf{z}$ denotes a signal (or time sequence) $\mathbf{z} = \{z(0), z(1), \cdots\}$, whose cardinality is inferred from the context. A signal that depends on a parameter θ is denoted as $\mathbf{z}(\theta)$, and $z(i;\theta)$ denotes its i-th element. A closed polyhedron $\mathcal{X} \subset \mathbb{R}^n$ is a set that results of the intersection of a finite number of hyperplanes as follows: $\mathcal{X} = \bigcap_i \{x : F_i x \leq f_i\}$, where $F_i \in \mathbb{R}^{1 \times n}$ and $f_i \in \mathbb{R}$.

Problem Statement

In this article, for the sake of simplicity, we consider that the system to be controlled can be modeled as a linear time-invariant system described by a discrete-time state-space linear model

$$x(k+1) = Ax(k) + Bu(k) \tag{1a}$$

$$y(k) = Cx(k), \tag{1b}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, and $y(k) \in \mathbb{R}^p$ are the state, the manipulable inputs, and the outputs of the system at time step k, respectively. This model will be used to calculate the predictions in the predictive controller.

The evolution of the plant must be such that the constraint

$$(x(k), u(k)) \in \mathcal{Z} \tag{2}$$

is satisfied for all $k \geq 0$. It is assumed that set $\mathcal{Z}$ is a closed polyhedron with a nonempty interior. Without loss of generality, it is assumed that $(0, 0) \in \mathcal{Z}$.

The main objective of tracking model predictive control is to stabilize the plant fulfilling the constraints and to steer the system output to the reference, that is, steer the tracking error $y - r$ to zero. In order to predict the expected evolution of the tracking error, some assumptions on the future values of the reference $\mathbf{r}$ must be considered. Since the reference may differ from expected, the tracking problem is inherently uncertain.

Thus, assuming that the reference signal is known a priori, $\mathbf{r} = \{r(0), r(1), \cdots\}$, the tracking model predictive control law $u(k) = \kappa(x(k), \mathbf{r})$ must be designed to ensure that the resulting controlled system

$$x(k+1) = Ax(k) + B\kappa(x(k), \mathbf{r})$$

$$y(k) = Cx(k)$$

satisfies the constraints, i.e., $(x(k), \kappa(x(k), \mathbf{r})) \in \mathcal{Z}$ for all $k \geq 0$, is stable, and, if it is possible, the controlled output converges to the reference, that is,

$$\lim_{k \to \infty} \|y(k) - r(k)\| = 0.$$

It is assumed that the system is stabilizable and that the outputs are linearly independent. It is also assumed that the state is measured and available at each sample time.

T

Tracking MPC for a Constant Reference

The most simple tracking problem is to consider that the reference signal is a constant signal in the future equal to the actual value of the reference, i.e., $r(k) = r$. This control problem is known as constant reference or setpoint tracking, and it is very common in the process industry, for instance, where processes are typically designed to operate at certain equilibrium point.

Determining the MPC Setpoint

Corresponding to each value of the reference, r, a MPC setpoint (x_r, u_r) must be calculated. This is an equilibrium point of the prediction model

$$x_r = Ax_r + Bu_r, \tag{3}$$

such that it guarantees a null tracking error

$$y_r = Cx_r = r. \tag{4}$$

Conditions for the existence of a setpoint possessing the above properties are given in Rawlings et al. (2017, Lemma 1.14).

The MPC setpoint must also fulfill the constraints (2)

$$(x_r, u_r) \in \mathcal{Z}.$$

In practice, the condition $(x_r, u_r) \in \mathcal{Z}$ is replaced by $(x_r, u_r) \in \mathcal{Z}_s$, where $\mathcal{Z}_s$ is a set contained in the interior of $\mathcal{Z}$. This allows us to ensure that the constraints $(x_r, u_r) \in \mathcal{Z}$ are not active once the system reaches the setpoint, which might lead to a loss of controllability.

For a given reference r, the MPC setpoint can be calculated by solving the following optimization problem:

$$(x_r, u_r) = \arg \min_{(x_s, u_s) \in \mathcal{Z}_s} \ell_t(x_s, u_s, r) \tag{5a}$$

$$s.t.\ x_s = Ax_s + Bu_s, \tag{5b}$$

where ℓ_t is a strictly convex function that penalizes the tracking error. It is typically chosen as the following quadratic function:

$$\ell_t(x_s, u_s, r) = \|Cx_s - r\|_{Q_s}^2 + \|u_s\|_{R_s}^2.$$

This problem is referred to as steady-state target optimization problem (Rao and Rawlings 1999).

Model Predictive Controller Design

If the reference to be tracked is a constant, i.e., $r(k) = r$ for all k, then the control objective is to stabilize the system and steer the initial state $x(0)$ to the MPC setpoint (x_r, u_r).

For a given reference r, first the MPC setpoint is calculated, and then the MPC is designed to regulate the system to this setpoint. Therefore, the MPC optimization problem depends on the current state x and the constant reference r (notice that the MPC setpoint is a function of the reference r), i.e., $P_N(x, r)$. Its solution gives an optimal control sequence $\mathbf{u}^o(x, r) = \{u^o(0; x, r), u^o(1; x, r), \cdots, u^o(N - 1; x, r)\}$ and an optimal state trajectory $\mathbf{x}^o(x, r) = \{x^o(0; x, r) = x, x^o(1; x, r), \cdots, x^o(N; x, r)\}$, where N is the prediction horizon. The first element of this sequence, namely, $u^o(0; x, r)$, is applied to the system.

In this chapter, a stabilizing design of the predictive controller based on a terminal cost and a terminal constraint is adopted. This is basically a regulation MPC penalizing the deviation w.r.t. the MPC setpoint. The optimization problem $P_N(x, r)$ is defined as follows:

$$\min_{\mathbf{u}} \sum_{j=0}^{N-1} \ell(x(j), u(j), r) + V_f(x(N), r)$$

$$s.t.\ x(0) = x, \tag{6a}$$

$$x(j+1) = Ax(j) + Bu(j), \quad j \in \mathbb{I}_{N-1} \tag{6b}$$

$$(x(j), u(j)) \in \mathcal{Z}, \quad j \in \mathbb{I}_{N-1} \tag{6c}$$

$$x(N) \in X_f(r). \tag{6d}$$

The stage cost function $\ell(\cdot)$ is a measure of the predicted tracking error , that is, $\ell(x_r, u_r, r) = 0$ and $\ell(x, u, r) \geq \alpha_1(\|x - x_r\|)$. The terminal cost function $V_f(\cdot)$ is such that

$$\alpha_2(\|x - x_r\|) \leq V_f(x_r, r) \leq \alpha_3(\|x - x_r\|).$$

The functions α_1, α_2, and α_3 are $\mathcal{K}_\infty$ functions. (A function $\alpha : [0, \infty) \to [0, \infty)$ is $\mathcal{K}_\infty$ if it is continuous, strictly increasing, $\alpha(0) = 0$ and $\lim_{s\to\infty} \alpha(s) = \infty$.)

The set of states where this optimization problem is feasible depends on the reference r, and it is denoted as $X_N(r)$. The solution of the optimal control problem $P_N(x, r)$ yields the receding horizon control law

$$\kappa_N(x, r) = u^o(0; x, r),$$

and the system controlled by this model predictive control is given by

$$x(k+1) = Ax(k) + B\kappa_N(x(k), r). \quad (7)$$

Because the horizon N is finite, x_r is not necessarily asymptotically stable for this system, but asymptotic stability can be ensured if the terminal cost function $V_f(\cdot)$ and the terminal region $X_f(r)$ are chosen appropriately. To this aim, the functions $\ell(\cdot)$, $V_f(\cdot)$ and the set $X_f(r)$ must satisfy the following condition:

Stability conditions for tracking MPC: For all $x \in X_f(r)$, there exists a control input u such that $(x, u) \in \mathcal{Z}$, the successor state $x^+ = Ax + Bu$ is contained in $X_f(r)$ and

$$V_f(x^+, r) - V_f(x, r) \le -\ell(x, u, r).$$

Under these stability conditions, the optimization problem is recursively feasible, i.e., if $P_N(x(0), r)$ is feasible, then all subsequent problems $P_N(x(i), r)$ are also feasible. Besides, the optimal cost function is a Lyapunov function of the system (7). Then, the MPC setpoint (x_r, u_r) is an asymptotically stable equilibrium point of the system (7), and the domain of attraction is $X_N(r)$ (Rawlings et al. 2017) A particular case when the stability conditions are trivially satisfied is taking $X_f(r) = x_r$ and $V_f(x, r) = 0$.

Tracking MPC for a Changing Reference

The previous predictive controller is inherently deterministic, since it is assumed that the reference is known, and this will remain constant in the future. However, in a realistic scenario, the reference may be changed without a predefined deterministic law. As it will be shown next, the changing references may make the stabilizing design of (6) fail. In this section, a tracking predictive controller for the case when the reference is piecewise constant (or varying, but ultimately converging to a constant value) is presented.

Feasibility and Stability Issues

If the reference r is constant, then tracking MPC (6) ensures asymptotic stability of the target state x_r and convergence to zero of the tracking error $(y - r)$. However, if the reference r varies, recursive feasibility (i.e., feasibility of $P_N(x(k), r(k))$ at each time instant k) and asymptotic stability may be compromised.

Consider that the tracking MPC has been designed to steer the system to the reference $r = r_1$ and the state x is in the domain of attraction $X_N(r_1)$. If the value of r changes from r_1 to r_2, the feasibility region, i.e., the domain of attraction of the controller, changes from $X_N(r_1)$ to $X_N(r_2)$. The current state x, which lies in $X_N(r_1)$, does not necessarily lie in $X_N(r_2)$ so that $\kappa_N(x, r_2)$ would be undefined and the model predictive controller would fail. Furthermore, the terminal constraint set $X_f(r_1)$ and the terminal cost function $V_f(\cdot, r_1)$ might not satisfy the stabilizing conditions of the tracking MPC (6) for the reference r_2 which would require their recomputation for the new reference.

This phenomenon is illustrated for the double integrator system, for which the matrices of model (1) are given by,

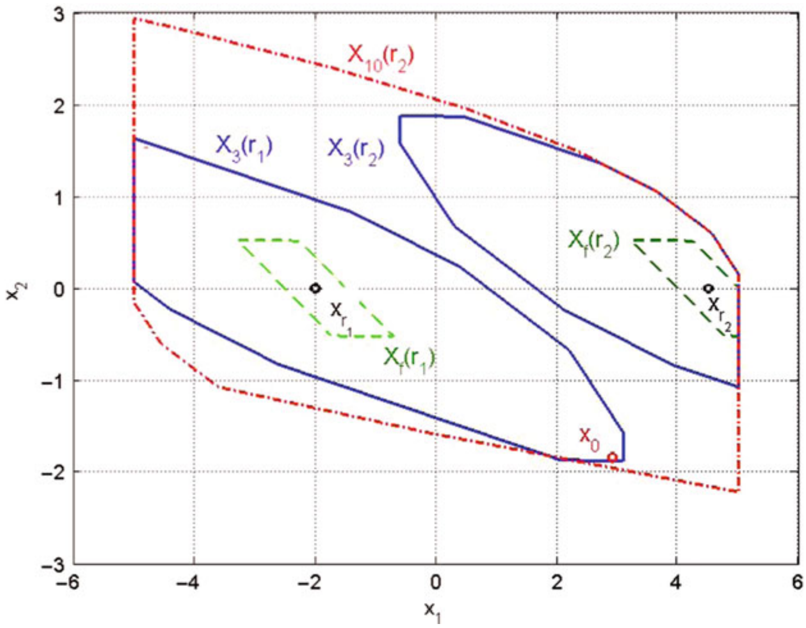
$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0.5 \\ 1 & 0.5 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

and the set of constraints is given by

$$\mathcal{Z} = \{(x, u) : \|x\|_\infty \le 5,\ \|u\|_\infty \le 0.3\}.$$

The initial state is $x(0) = (2.91, -1.83)$ and the initial value of the reference is $r_1 = -2$. The corresponding MPC setpoint is (x_{r_1}, u_{r_1}) where $x_{r_1} = (-2, 0)$ and $u_{r_1} = (0, 0)$. If the reference

T



Tracking Model Predictive Control, Fig. 1 Example of the double integrator: terminal regions ($X_f(r_1)$ and $X_f(r_2)$) and domains of attraction of MPC ($X_3(r_1)$, $X_3(r_2)$, and $X_{10}(r_2)$)

changes from $r_1 = -2$ to $r_2 = 4.5$, then the new MPC setpoint is (x_{r_2}, u_{r_2}) where $x_{r_2} = (4.5, 0)$ and $u_{r_2} = (0, 0)$. The horizon is chosen to be $N = 3$, and the domains of attraction for the two values of r are, respectively, $X_3(r_1)$ and $X_3(r_2)$. As it is shown in Fig. 1, these two domains, $X_3(r_1)$ and $X_3(r_2)$, are disjoint. While $r = r_1$, the state trajectory commencing at $x(0) \in X_3(r_1)$ remains in $X_3(r_1)$. If r subsequently changes its value to r_2 at the sample k_1, the model predictive controller fails since $x(k_1)$ does not lie in $X_3(r_2)$.

These feasibility and stability issues can be overcome if the predictive controller is redesigned for the new setpoint. This would require the calculation of a new terminal set and a prediction horizon each time the setpoint changes. For instance, in the example of Fig. 1, if the terminal constraint is recalculated for r_2 and the prediction horizon is chosen as $N = 10$, then when the reference r changes from r_1 to r_2, the MPC controller to be used is $\kappa_{10}(\cdot, r_2)$, which steers the system to the reference r_2 since $x(0) \in X_3(r_1) \subset X_{10}(r_2)$. This recalculation can be done off-line if the MPC setpoint changes are known a priori (Findeisen et al. 2000; Wan and Kothare 2003). Other methods to avoid this issues are: design a predictive controller to provide a certain degree of robustness to setpoint variations (Pannocchia and Kerrigan 2005; Pannocchia 2004), a predictive control law with a mode to recover recursive feasibility (Rossiter et al. 1996; Chisci and Zappa 2003), or using specialized predictive control laws (Magni et al. 2001; Magni and Scattolini 2005). Another solution is to use a reference governor and a predictive controller (Bemporad et al. 1997; Olaru and Dumur 2005).

Stabilizing MPC for Tracking

The idea behind the reference governor is to introduce an artificial reference r^a that is manipulated to ensure that the current state is in the domain of attraction $X_N(r^a)$ and which tends to the actual reference r if r remains constant or tends to a constant. In Limon et al. (2008), this idea is used to formulate the so-called MPC for tracking. The artificial reference r^a is an extra decision variable in the optimal control problem that avoids the loss of feasibility issue. In order to enforce the convergence to the actual reference r, a term that penalizes the deviation between the artificial reference r^a and the actual reference r, $\ell_o(r^a, r)$ is added. This function is assumed to be convex in r^a. A suitable choice of this term is the cost function of the steady-state target calculator (5), i.e., $\ell_o(r^a, r) = \ell_t(x_{r^a}, u_{r^a}, r)$, where

(x_{r^a}, u_{r^a}) is the artificial setpoint associated to the artificial reference r^a.

The optimal control problem $P_N^t(x, r)$ of the MPC for tracking is given by

$$\min_{\mathbf{u}, r^a} \sum_{j=0}^{N-1} \ell(x(j), u(j), r^a) + V_f(x(N), r^a) + \ell_o(r^a, r)$$

$$s.t.\ x(0) = x, \tag{8a}$$

$$x(j+1) = Ax(j) + Bu(j), \quad j \in \mathbb{I}_{N-1} \tag{8b}$$

$$(x(j), u(j)) \in \mathcal{Z}, \quad j \in \mathbb{I}_{N-1} \tag{8c}$$

$$r^a \in \mathcal{R} \tag{8d}$$

$$(x(N), r^a) \in \Gamma, \tag{8e}$$

where

$$\mathcal{R} = \{r : (x_r, u_r) \in \mathcal{Z}_s, \; Ax_r + Bu_r = x_r,\ Cx_r = r\}. \tag{9}$$

Condition (8e) is an extended terminal constraint of both, the terminal state $x(N)$ and the artificial reference r^a. The feasibility region of this optimization problem X_N^t is the set of states that can be steered to any reference of the set $\mathcal{R}$ in N steps, that is,

$$X_N^t = \bigcup_{r^a \in \mathcal{R}} X_N(r^a).$$

The terminal cost function $V_f(\cdot)$ and the terminal constraint set, Γ, must satisfy appropriately modified stability conditions in order to ensure recursive feasibility and asymptotic stability of (x_r, u_r). The stability conditions are:

Stability conditions of MPC for tracking: For all $(x, r^a) \in \Gamma$: $r^a \in \mathcal{R}$, and there exist a u satisfying: (i) $(x, u) \in \mathcal{Z}$, (ii) the successor state $x^+ = Ax + Bu$ is such that $(x^+, r^a) \in \Gamma$ and $V_f(x^+, r^a) - V_f(x, r^a) \leq -\ell(x, u, r^a)$.

As shown in Limon et al. (2008), if the control action u of the former stability conditions is given by the following so-called terminal control law $u = K(x - x_{r^a}) + u_{r^a}$ with K such that the eigenvalues of $A + BK$ are in the unitary disk, then the terminal set Γ is a polyhedron that can be computed using standard algorithms to compute positively invariant sets for constrained linear systems. A simple choice of the terminal cost and constraint satisfying these stability conditions is $V_f(\cdot) = 0$ and $\Gamma = \{(x, r^a) : x = x_{r^a}\}$.

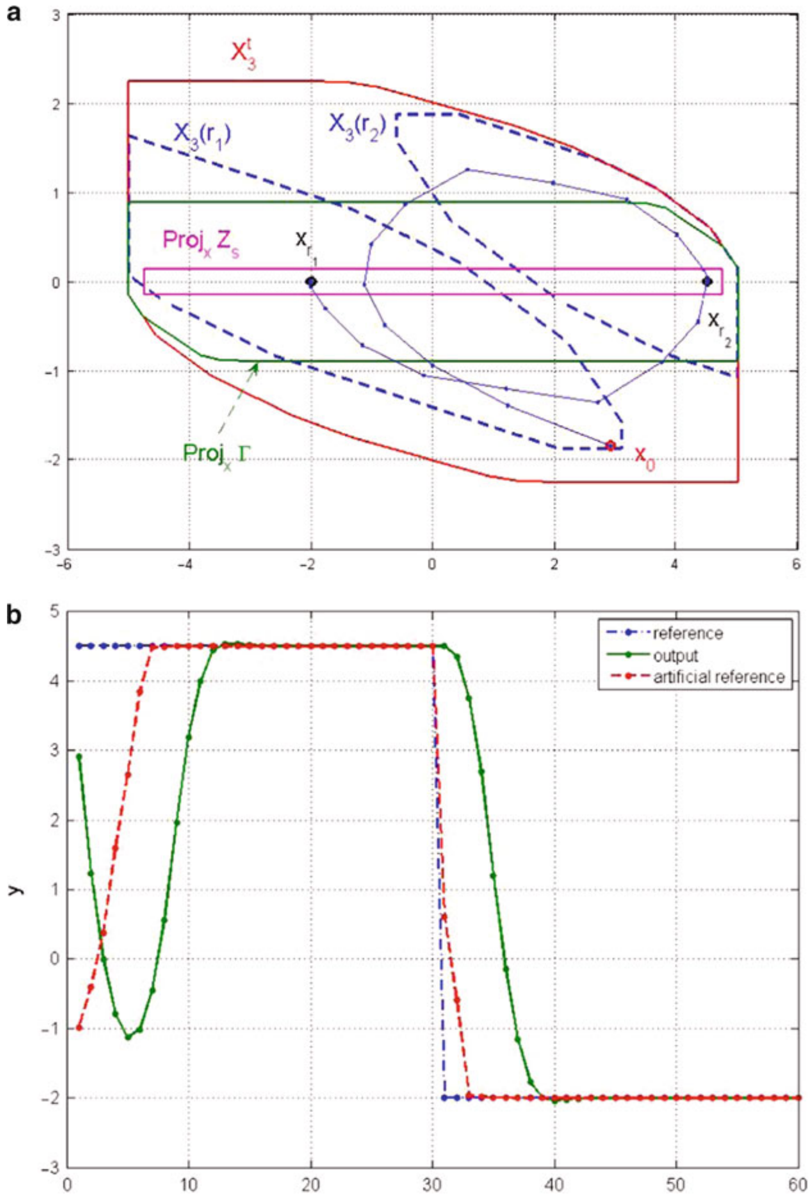
Theorem 1 *If the stability conditions of the MPC for tracking hold, then the predictive control law derived from the optimal control problem $P_N^t(x, r)$ is such that:*

(1) *For every feasible initial state, i.e., $x(0) \in X_N^t$ and for all $r \in \mathbb{R}^p$, the optimization problem is recursively feasible, that is, if $P_N^t(x(0), r)$ is feasible, then all the subsequent problems $P_N^t(x(i), r)$ will also be feasible.*
(2) *If r is admissible, i.e., $r \in \mathcal{R}$, then the setpoint (x_r, u_r) is an asymptotically stable equilibrium point of the closed-loop system, and the domain of attraction is X_N^t.*
(3) *If r is not admissible, that is, $r \notin \mathcal{R}$, then the setpoint $(x_{r^\star}, u_{r^\star})$ such that*

$$r^\star = \arg \min_{r^a \in \mathcal{R}} \ell_o(r^a, r)$$

is asymptotically stable, and the domain of attraction is X_N^t.
(4) *The domain of attraction X_N^t is larger than or equal to the domain of the tracking MPC for any reference $r \in \mathcal{R}$, that is, $X_N(r) \subseteq X_N^t$, and contains all the equilibrium points that lie in $\mathcal{Z}_s$.*
(5) *If the reference $r(k)$ is not constant and converges to an steady value r, the optimization problem is recursively feasible, and the setpoint (x_{r^*}, u_{r^*}) is an asymptotically stable equilibrium point for all $x(0) \in X_N^t$.*

In Fig. 2 a, the aforementioned properties are illustrated for the example of the double integrator. The MPC for tracking has been designed with the same prediction horizon $N = 3$ and the same terminal control law and terminal cost function as in the previous tracking MPC case. The initial state is also the same, and the reference signal is $r(k) = r_2$ for $k \leq 30$ and $r(k) = r_1$ for



Tracking Model Predictive Control, Fig. 2 The double integrator controlled by the MPC for tracking. (**a**) Comparison of the domains of attraction of the tracking MPC $X_3(r_1)$ and $X_3(r_2)$ vs. the domain of attraction of the MPC for tracking X_3^t. (**b**) Trajectories of the reference, the controlled output, and the artificial reference r^a

$k > 30$. Notice that the tracking MPC cannot be used to do this without redesign. In Fig. 2 a, it can be seen that the domain of attraction of the MPC for tracking X_3^t is larger than the domain provided by the standard tracking MPC, $X_3(r_1)$, or $X_3(r_2)$. This figure also shows the state portrait of the closed-loop trajectory. In Fig. 2 b, the trajectories of the reference signal $\mathbf{r}$, the controlled output y, and the artificial reference r^a are depicted. Notice the role of the artificial reference: r^a differs from the reference in order to guarantee recursive feasibility and finally converges to the reference r to enforce asymptotic stability.

The stabilizing MPC for tracking has been extended to the case of nonlinear prediction models in Limon et al. (2018). This controller inherits the properties of its linear prediction model counterpart presented in this section.

Offset-Free Tracking

In practice, there may exist mismatches between the prediction model and the dynamics of the real plant to be controlled, due, for instance, to un-modeled nonlinearities or unmeasured disturbances. This would require to design the predictive controller to be robust to these uncertain effects. Even in the case that the predictive controller based on nominal predictions makes that the uncertain controlled system is stable and converges to a steady state, there may exist an steady error between reference and the output.

This offset can be canceled taking into account a prediction model corrected by a disturbance model (Pannocchia and Rawlings 2003; Maeder and Morari 2010). To achieve offset-free control, the disturbance is assumed to be constant, leading to the following augmented model:

$$x(k+1) = Ax(k) + Bu(k) + B_d d(k), \quad (10a)$$

$$d(k+1) = d(k), \quad (10b)$$

$$y(k) = Cx(k) + D_d d(k). \quad (10c)$$

Matrices B_d and D_d define the disturbance model and are typically chosen as $B_d = 0$ and $D_d = I_p$ (Pannocchia and Rawlings 2003).

The state $\hat{x}$ and the disturbance signal $\hat{d}$ are estimated using an observer based on the disturbance model (10). The disturbance model and the estimator gains can be calculated separately, but this may lead to a poor closed-loop performance. A joint design procedure has been proposed in Pannocchia and Bemporad (2007).

Once the estimated state and disturbance, $(\hat{x}, \hat{d})$ are available, the corrected prediction model

$$x(k+1) = Ax(k) + Bu(k) + B_d\hat{d}, \quad (11a)$$

$$y(k) = Cx(k) + D_d\hat{d}, \quad (11b)$$

must be used to replace the nominal prediction model in the steady-state target optimizer (5b) and in the tracking MPC (6b). In the MPC for tracking, the prediction model in (8b) and in (9) must be replaced by (11).

Future Directions

Tracking model predictive control is an inherently uncertain control problem due to the unexpected changes in the reference. Constant reference tracking has been widely studied, and there exist a number of nice solutions.

The case of trajectory tracking is not as mature as the setpoint tracking case. If the reference signal is known a priori, this can be used to design a suitable tracking predictive controller. This control problem can be solved by using a two-layer structure: a trajectory planning, aimed to calculate the optimal reachable trajectory, on top of a predictive control law that steers the system to the optimal reference. Asymptotic stability to the target trajectory can be achieved resorting to a regulation problem, using a terminal equality constraint, for instance. Another interesting line is to assume that the reference is the output of a certain dynamic system. For different families of trajectories, such as ramps or sinusoidal signals, Maeder (Maeder and Morari 2010) has proposed a reference tracking MPC based on extended disturbance models.

The problem of tracking MPC in case of unknown (or changing) reference signals is challenging and can be considered an open problem that deserves more research efforts. The MPC for tracking has been recently extended to deal with periodic references (Limon et al. 2016). This controller can asymptotically stabilize the system to the best reachable periodic trajectory even when the reference is changed. The only required assumption on the reference is that its period is known. The extension to cope with a more general class of references is an open problem.

Another interesting control problem is the tracking of unreachable (equilibrium point as well as trajectory) targets. Recently this problem has been posed as an economic model predictive control problem (Rawlings et al. 2017). Therefore, the stabilizing design of economic MPC (Angeli et al. 2012) can be extended to the case of tracking unreachable targets.

Cross-References

- ▶ Economic Model Predictive Control
- ▶ Nominal Model-Predictive Control
- ▶ Regulation and Tracking of Nonlinear Systems
- ▶ Tracking and Regulation in Linear Systems

Recommended Reading

The book (Camacho and Bordons 2004) covers the classic approach to the tracking MPC. In Rawlings et al. (2017), the authors deal with the tracking MPC in a very general and clear way and survey existing results on stability, target calculation, and offset-free control for linear and nonlinear models. In Muske (1997), the reachability of setpoints is studied and in Rao and Rawlings (1999) the target calculation problem. Disturbance models are widely analyzed in Pannocchia and Rawlings (2003), Pannocchia and Bemporad (2007), Maeder et al. (2009), and Maeder and Morari (2010). Another offset-free MPC based on the internal model principle can be found in Magni and Scattolini (2007). Further results on MPC for tracking are addressed in Ferramosca et al. (2009) and Limon et al. (2016, 2018). A survey on the MPC for tracking can be found in Limon et al. (2012). The authors have also extended the MPC for tracking to the robust case both in constant reference and periodic reference framework (Pereira et al. (2017)).

Acknowledgments The authors would like to thank the support received by the MINECO-Spain and FEDER Funds under project DPI2016-76493-C3-1-R.

Bibliography

Angeli D, Amrit R, Rawlings JB (2012) On average performance and stability of economic model predictive control. IEEE Trans Autom Control 57:1615–1626

Bemporad A, Casavola A, Mosca E (1997) Nonlinear control of constrained linear systems via predictive reference management. IEEE Trans Autom Control 42:340–349

Camacho EF, Bordons C (2004) Model predictive control, 2nd edn. Springer, New York

Chisci L, Zappa G (2003) Dual mode predictive tracking of piecewise constant references for constrained linear systems. Int J Control 76:61–72

Ferramosca A, Limon D, Alvarado I, Alamo T, Camacho EF (2009) MPC for tracking with optimal closed-loop performance. Automatica 45:1975–1978

Findeisen R, Chen H, Allgöwer F (2000) Nonlinear predictive control for setpoint families. In: Proceedings of the American control conference

Limon D, Alvarado I, Alamo T, Camacho EF (2008) MPC for tracking of piece-wise constant references for constrained linear systems. Automatica 44:2382–2387

Limon D, Ferramosca A, Alamo T, Gonzalez AH (2012) Model predictive control for changing economic targets. Paper presented at the IFAC conference on nonlinear model predictive control 2012 (NMPC'12). Noordwijkerhout, 23–27 Aug 2012

Limon D, Pereira M, Muñoz de la Peña D, Alamo T, Jones CN, Zeillinger MN (2016) MPC for tracking periodic references. IEEE Trans Autom Control 61:1123–1128

Limon D, Ferramosca A, Alvarado I, Alamo T (2018) Nonlinear MPC for tracking piece-wise constant reference signals. IEEE Trans Autom Control 63:23735–3750

Maeder U, Morari M (2010) Offset-free reference tracking with model predictive control. Automatica 46(9):1469–1476

Maeder U, Borrelli F, Morari M (2009) Linear offset-free model predictive control. Automatica 45:2214–2222

Magni L, Scattolini R (2005) On the solution of the tracking problem for non-linear systems with MPC. Int J Syst Sci 36(8):477–484

Magni L, Scattolini R (2007) Tracking on non-square nonlinear continuous time systems with piecewise constant model predictive control. J Process Control 17: 631–640

Magni L, De Nicolao G, Scattolini R (2001) Output feedback and tracking of nonlinear systems with model predictive control. Automatica 37:1601–1607

Muske K (1997) Steady-state target optimization in linear model predictive control. Paper presented at the 16th American control conference, Alburquerque, 4–6 June 1997

Olaru S, Dumur D (2005) Compact explicit MPC with guarantee of feasibility for tracking. In: Conference decision and control and European control conference 2005, pp 969–974

Pannocchia G (2004) Robust model predictive control with guaranteed setpoint tracking. J Process Control 14:927–937

Pannocchia G, Bemporad A (2007) Combined design of disturbance model and observer for offset-free model predictive control. IEEE Trans Autom Control 52:1048–1053

Pannocchia G, Kerrigan E (2005) Offset-free receding horizon control of constrained linear systems. AIChE J 51:3134–3146

Pannocchia G, Rawlings JB (2003) Disturbance models for offset-free model-predictive control. AIChE J 49:426–437

Pereira M, Muñoz de la Peña D, Limon D, Alvarado I, Alamo T (2017) Robust model predictive controller for tracking changing periodic signals. IEEE Trans Autom Control 62:5343–5350
Rao CV, Rawlings JB (1999) Steady states and constraints in model predictive control. AIChE J 45:1266–1278
Rawlings JB, Mayne DQ, Diehl MM (2017) Model predictive control: theory, computation and design, 2nd edn. Nob-Hill Publishing, Madison
Rossiter JA, Kouvaritakis B, Gossner JR (1996) Guaranteeing feasibility in constrained stable generalized predictive control. IEEE Proc Control Theory Appl 143:463–469
Wan Z, Kothare MV (2003) An efficient off-line formulation of robust model predictive control using linear matrix inequalities. Automatica 39:837–846

Trajectory Generation for Aerial Multicopters

Mark W. Mueller[1] and Raffaello D'Andrea[2]
[1]University of California, Berkeley, CA, USA
[2]ETH Zurich, Zurich, Switzerland

Abstract

The ability to generate feasible and safe trajectories is crucial for autonomous multicopter systems. These trajectories can ideally be generated on low-cost, embedded computational hardware and exploit the system's full dynamic capabilities while satisfying constraints. As operations increasingly focus on operation at high speeds, or in dynamically changing environments, strategies are required that can rapidly plan and replan trajectories. This entry reviews typical approaches for trajectory generation of aerial robots, with a focus on multicopters, and discusses various approximations that may be used to make the problem more tractable. The strategy of planning in higher derivatives of the vehicle position (such as acceleration, jerk, and snap) is discussed in depth. We also discuss the related issue of expressing system limitations and constraints in these derivatives. Finally, possible future directions are discussed.

Keywords

Multicopters · UAV · Planning · Differential flatness

Introduction

Aerial robots are increasingly routinely used in everyday operations, accomplishing tasks such as remote sensing, surveillance, and delivery of goods and people. Continuing technological development, especially of energy storage, and development of the regulatory environment mean that such systems may soon be commonplace. A crucial requirement for their autonomous operations is the ability to plan motions to achieve goals, especially trajectories moving them through space. Due to their mechanical simplicity, and the ability to hover in place when required, the most common aerial robots are multicopters, especially quadcopters.

A typical trajectory generation problem consists of moving a single multicopter from an initial state (typically described as a position, velocity, orientation, and angular velocity) to a final state. The final state may be as detailed as the initial state, or it may only specify some components (such as an emergency stopping trajectory that requires simply zero final velocity). The resulting trajectory must respect the system dynamics and potentially avoid additional constraints, such as collisions. The satisfaction of system dynamics and constraints is to be understood as satisfying some sufficiently accurate approximation of the true system capabilities (i.e., sufficiently accurate for the problem at hand). Moreover, the available resources (e.g., computational power and computation time) are typically important considerations, as is the ability to accurately model the system and disturbances.

This entry focuses on model-based approaches, that is, generation methods that rely on first-principle models of the multicopter systems to generate trajectories. An additional focus is on low computational complexity, so that trajectories may be computed on constrained hardware.