22 January 2009, Grenoble, France

A flatness-based iterative method for reference trajectory generation in constrained NMPC

J.A. De Doná [1,2], F. Suryawan [1], M.M. Seron [1], J. Lévine [2]

(1) CDSC, The University of Newcastle, Australia

(2) CAS, ENSMP, Fontainebleau, France

CDSC

ECOLE DES MINES
DE PARIS

Differentially Flat Systems

Consider a general nonlinear system:

$$\dot{x}(t) = f\big(x(t), u(t)\big)$$

where $x(t) \in I\!R^n$, $u(t) \in I\!R^m$.

If the system is flat we can write all trajectories $(x(t), u(t))$ in terms of a finite set of variables, $y(t) \in I\!R^m$ (known as the flat outputs):

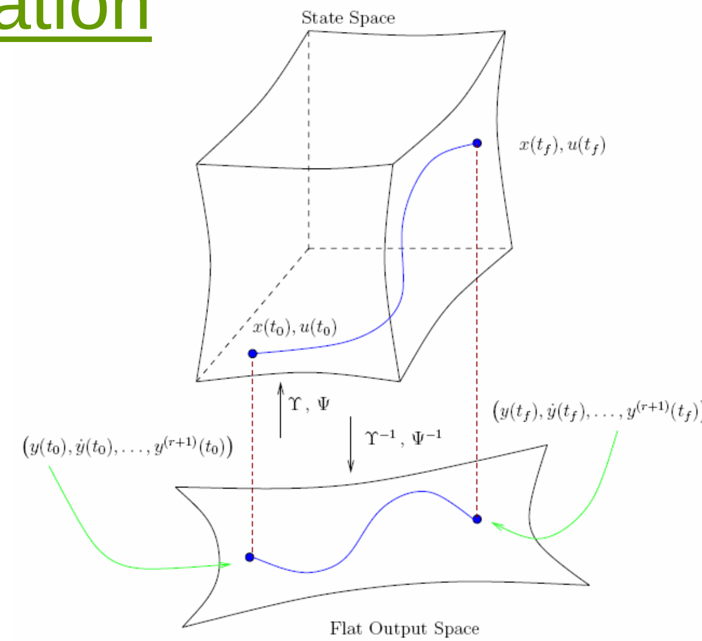
$$x(t) = \Upsilon\big(y(t), \dot{y}(t), \ddot{y}(t), \ldots, y^{(r)}(t)\big),$$
$$u(t) = \Psi\big(y(t), \dot{y}(t), \ddot{y}(t), \ldots, y^{(r+1)}(t)\big).$$

For simplicity, we assume that the flat output is given simply as a (nonlinear) combination of the states:

$$y = h(x)$$

(This is not the most general case, $y = \Phi(x, u, \dot{u}, \ddot{u}, \ldots, u^{(p)})$, but it represents many cases of practical interest.)

Reference Trajectory Generation



Goal:

Prescribe a 'desired' reference trajectory for the flat output, $y^{\text{ref}}(t)$, and obtain the corresponding input and state trajectories:

$$\begin{cases} u^{\text{ref}}(t) = \Psi\left(y^{\text{ref}}(t), \dot{y}^{\text{ref}}(t), \ldots, (y^{\text{ref}}(t))^{(r+1)}\right), \\ x^{\text{ref}}(t) = \Upsilon\left(y^{\text{ref}}(t), \ldots, (y^{\text{ref}}(t))^{(r)}\right) \end{cases}$$

Challenge:

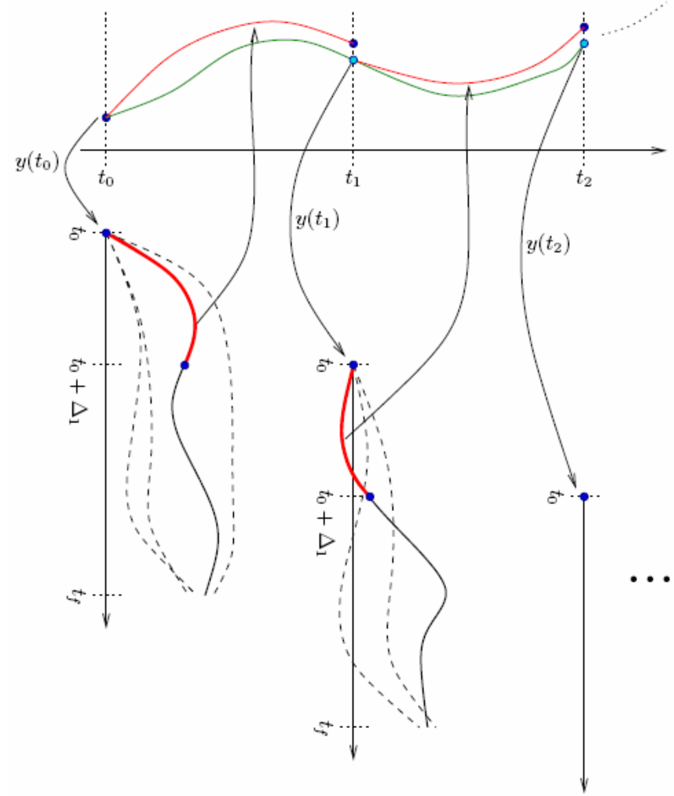
Constraints:

$$u(t) \in \mathbb{U},\ x(t) \in \mathbb{X}$$

Iterative Method:

- Parametrise $y^{\text{ref}}(t)$ with splines (performance objectives).
- Evaluate constraints using MPC for trajectory tracking.
- Reshape $y^{\text{ref}}(t)$ based on the "feedback from present and future constraints."

The method can be applied in a receding horizon fashion:



Parameterisation of flat outputs using splines

We consider the problem of steering the system from an initial state at time t_0 to a final state at time t_f. We parameterise the flat outputs $y_j(t),\ j = 1, \ldots, m,$

$$y_j(t) = \sum_{i=1}^{N} \lambda_i(t) P_{ij}; \qquad t \in [t_0\ ,\ t_f],$$

where $\lambda_i \in \mathcal{C}^{r+1}[t_0, t_f], i = 1, \ldots, N,$ is a set of basis functions.

Discretise at a set of M+1 sampling times, $t_0, t_1, \ldots, t_f$

$$Y_j \triangleq \begin{bmatrix} y_j(t_0), \\ y_j(t_1), \\ \vdots \\ y_j(t_f) \end{bmatrix} = G_0 \begin{bmatrix} P_{1j}, \\ \vdots \\ P_{Nj} \end{bmatrix} = G_0 P_j,$$

where

$$G_0 \triangleq \begin{bmatrix} \lambda_1(t_0) & \ldots & \lambda_N(t_0) \\ \vdots & \ddots & \vdots \\ \lambda_1(t_f) & \ldots & \lambda_N(t_f) \end{bmatrix}$$

is the *basis function matrix* (also known as *blending matrix*).

Collecting all the *m* flat outputs,

$$Y \triangleq \begin{bmatrix} Y_1 & Y_2 & \ldots & Y_m \end{bmatrix}$$

$$= \begin{bmatrix} y_1(t_0) & y_2(t_0) & \ldots & y_m(t_0) \\ \vdots & \vdots & \ddots & \vdots \\ y_1(t_f) & y_2(t_f) & \ldots & y_m(t_f) \end{bmatrix}$$

$$= G_0 \cdot \begin{bmatrix} P_1 & P_2 & \ldots P_m \end{bmatrix} = G_0\, P = Y(P)$$

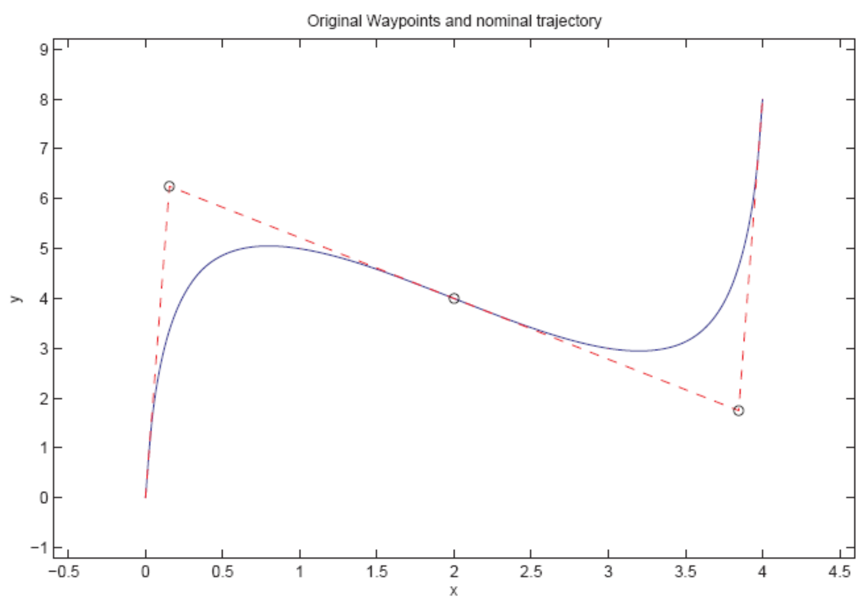
Similarly for the flat output derivatives:

$$Y^{(1)} = G_1 P\,; \quad Y^{(2)} = G_2 P\,; \quad Y^{(3)} = G_3 P\,; \quad \ldots \quad Y^{(r+1)} = G_{r+1} P\,;$$

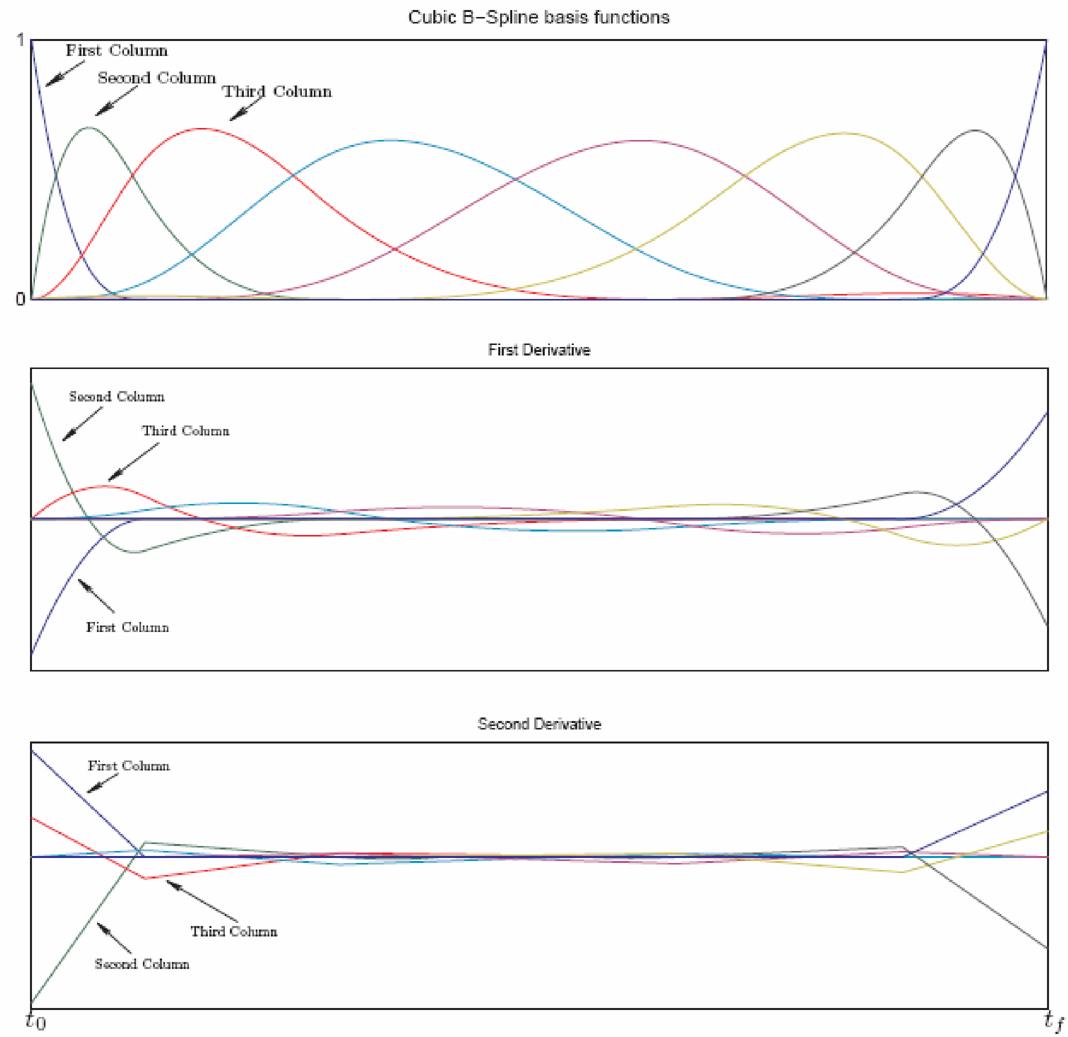
where $Y^{(q)} \triangleq [Y_1^{(q)}\ Y_2^{(q)}\ \ldots\ Y_m^{(q)}]$, and

$$Y_j^{(q)} \triangleq \begin{bmatrix} \frac{d^q}{dt^q} y_j(t)\Big|_{t=t_0} \\ \vdots \\ \frac{d^q}{dt^q} y_j(t)\Big|_{t=t_f} \end{bmatrix} ; \qquad G_q \triangleq \begin{bmatrix} \frac{d^q}{dt^q}\lambda_1(t)\Big|_{t=t_0} & \cdots & \frac{d^q}{dt^q}\lambda_N(t)\Big|_{t=t_0} \\ \vdots & \ddots & \vdots \\ \frac{d^q}{dt^q}\lambda_1(t)\Big|_{t=t_f} & \cdots & \frac{d^q}{dt^q}\lambda_N(t)\Big|_{t=t_f} \end{bmatrix}$$

The rows of the *N* x *m* matrix *P* are *m*-dimensional vectors called ***control points***:



In this work we have used ‘Clamped B-splines,’ which results in G_0 , G_1 , … having a particular structure:



Columns of G_0

Columns of G_1

Columns of G_2

From:

$$\frac{d^q}{dt^q} y_j(t)\Big|_{t=t_0} = \sum_{i=1}^{N} \frac{d^q}{dt^q} \lambda_i(t)\Big|_{t=t_0} P_{ij},$$

we can see that, e.g., prescribed position, first and second derivatives can be maintained fixed at t_0 and t_f (*rest-to-rest*) by hoding the external control points (three top and three bottom rows of *P*) fixed.

Given an initial reference trajectory and the corresponding control points P^{ref}, the idea is to change the internal control points of P^{ref} to shape the trajectory. We reparameterise the control points as:

$$P = P^{\mathrm{ref}} + \rho \hat{P}; \qquad \rho = \begin{pmatrix} 0 \\ I \\ 0 \end{pmatrix}$$

where matrix $\hat{P}$ parameterises the deviation from the 'internal' control points of P^{ref}

Using MPC to shape the reference trajectory

Given a specified reference trajectory for the flat outputs, parameterised by reference control points P^{ref}, for j=1,…,m:

$$y_j^{\text{ref}}(t) = \sum_{i=1}^{N} \lambda_i(t) P_{ij}^{\text{ref}}; \quad t \in [t_0, t_f]$$

From flatness property, obtain input and state reference trajectories

$$\left(u^{\text{ref}}(t), x^{\text{ref}}(t), y^{\text{ref}}(t)\right)$$

Linearise around reference trajectory

$$\dot{\tilde{x}}(t) = A(t)\tilde{x}(t) + B(t)\tilde{u}(t),$$
$$\tilde{y}(t) = C(t)\tilde{x}(t)$$

$$\begin{cases} \tilde{u}(t) \triangleq u(t) - u^{\text{ref}}(t), \\ \tilde{x}(t) \triangleq x(t) - x^{\text{ref}}(t), \\ \tilde{y}(t) \triangleq y(t) - y^{\text{ref}}(t), \\ A(t) = \left(\partial f / \partial x\right)\Big|_{x^{\text{ref}}(t), u^{\text{ref}}(t)}, \\ B(t) = \left(\partial f / \partial u\right)\Big|_{x^{\text{ref}}(t), u^{\text{ref}}(t)}, \\ C(t) = \left(\partial h / \partial x\right)\Big|_{x^{\text{ref}}(t)}. \end{cases}$$

Discretise with sampling interval $T_s \triangleq (t_f - t_0)/M$

$$\tilde{x}_{k+1} = A_k\tilde{x}_k + B_k\tilde{u}_k,$$
$$\tilde{y}_k = C_k\tilde{x}_k$$

(In vectorised form: $\tilde{\mathbf{x}} = \Gamma\tilde{\mathbf{u}} + \Omega\tilde{x}_0$, $\tilde{Y}_j = \mathbf{C}_j \begin{bmatrix} \tilde{x}_0 \\ \Gamma\tilde{\mathbf{u}} + \Omega\tilde{x}_0 \end{bmatrix}$.)

↓

Formulate performance objective of deviation system as a regulation problem to the origin

$$V_M(\{\tilde{x}_k\}, \{\tilde{u}_k\}, \{\tilde{y}_k\}) \triangleq \frac{1}{2}\tilde{x}_M^{\mathrm{T}} P \tilde{x}_M + \frac{1}{2}\sum_{k=0}^{M-1} \tilde{y}_k^{\mathrm{T}} Q \tilde{y}_k + \frac{1}{2}\sum_{k=0}^{M-1} \tilde{u}_k^{\mathrm{T}} R \tilde{u}_k$$

with $\tilde{x}_0 \triangleq x(t) - x^{\mathrm{ref}}(t_0)$, $x(t)$ = current state.

↓

Obtain the optimal input sequence subject to constraints

$$\tilde{\mathbf{u}}^{\mathrm{opt}} = \begin{bmatrix} \tilde{u}_0^{\mathrm{opt}}, \\ \vdots \\ \tilde{u}_{M-1}^{\mathrm{opt}} \end{bmatrix} \triangleq \arg\min_{\tilde{\mathbf{u}}} \frac{1}{2}\tilde{\mathbf{u}}^{\mathsf{T}} H \tilde{\mathbf{u}} + \tilde{\mathbf{u}}^{\mathsf{T}} F \tilde{x}_0$$

subject to
$$|\mathbf{u}^{\mathrm{ref}} + \tilde{\mathbf{u}}| \le U_{\max}$$

(We considered, for simplicity, input constraints $|u(t)| \le u_{\max}$. State and output constraints can be included in a straightforward manner.)

↓

Compute the flat output trajectory obtained with MPC

↓

$$Y_j^{\mathsf{mpc}} \triangleq \mathbf{C}_j \begin{bmatrix} \tilde{x}_0 \\ \Gamma \tilde{\mathbf{u}}^{\mathsf{opt}} + \Omega \tilde{x}_0 \end{bmatrix} + Y_j^{\mathsf{ref}}$$

$\Downarrow$

Compute variation of internal control points $\hat{P}_j^{\mathsf{mpc}}$ that solves $G_0\left(P_j^{\mathsf{ref}} + \rho \hat{P}_j^{\mathsf{mpc}}\right) = Y_j^{\mathsf{mpc}}$ (in a least-squares sense)

$$\hat{P}_j^{\mathsf{mpc}} = \left((G_0\,\rho)^{\mathsf{T}} G_0\,\rho\right)^{-1} \left(G_0\,\rho\right)^{\mathsf{T}} \left(Y_j^{\mathsf{mpc}} - G_0 P_j^{\mathsf{ref}}\right)$$

Iterative method for ref. trajectory generation

Iterative Algorithm

Start with initial control points $P^{\text{ref},0}$ that parameterise an initial reference trajectory $Y^{\text{ref},0} = G_0 P^{\text{ref},0}$ (generated based on performance considerations);

Step 1. Given a set of control points $P^{\text{ref,k}}$;

Step 2. Compute $Y^{\text{ref,k}} = G_0 P^{\text{ref,k}}$;

Step 3. Compute $Y_j^{\text{mpc,k}}$, in general a nonlinear function of $P^{\text{ref,k}}$, $Y^{\text{mpc,k}} = G\big(P^{\text{ref,k}}\big)$;

Step 4. Find the variation of the control points that gives a reference trajectory that is closest to $Y^{\mathrm{mpc,k}}$:

$$\hat{P}_j^{\mathrm{mpc,k}} = \left((G_0\,\rho)^{\mathsf{T}} G_0\,\rho\right)^{-1} \left(G_0\,\rho\right)^{\mathsf{T}} \left(Y_j^{\mathrm{mpc,k}} - G_0 P_j^{\mathrm{ref,k}}\right).$$

Step 5. Update the control points according to:
$P^{\mathrm{ref,k+1}} = P^{\mathrm{ref,k}} + \rho \hat{P}^{\mathrm{mpc,k}}$.

Step 6. While (a weighted 2-norm of) the difference $\left(P^{\mathrm{ref,k+1}} - P^{\mathrm{ref,k}}\right)$ is larger than a prescribed tolerance level and the maximum number of iterations is not reached:
assign $P^{\mathrm{ref,k}} \leftarrow P^{\mathrm{ref,k+1}}$ and go to Step 1.

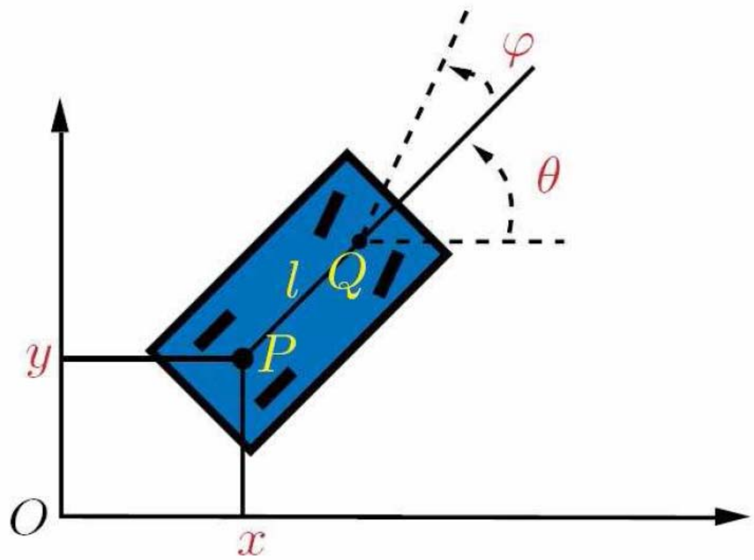
Note from Steps 1–5 that the proposed algorithm implements a recursion:

$$P^{\mathsf{ref,k+1}} = F\Big(P^{\mathsf{ref,k}}\Big)$$

whose complexity depends predominantly on the (in general, nonlinear) mapping $Y^{\mathsf{mpc,k}} = G\Big(P^{\mathsf{ref,k}}\Big)$.

The convergence properties of this recursive mapping, will be investigated in future work.

Simulation Example



Kinematics

$$\dot{x} = u\cos\theta;$$

$$\dot{y} = u\sin\theta;$$

$$\dot{\theta} = \frac{u}{l}\tan\varphi$$

The system's state and input can be completely determined from the flat output and its derivatives up finite order:

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} = \begin{pmatrix} x \\ y \\ \arctan(\frac{\dot{y}}{\dot{x}}) \end{pmatrix}$$

$$\mathbf{u} = \begin{pmatrix} u \\ \varphi \end{pmatrix} = \begin{pmatrix} \sqrt{\dot{x}^2 + \dot{y}^2} \\ \arctan\left(l\frac{\ddot{y}\dot{x} - \dot{y}\ddot{x}}{(\dot{x}^2+\dot{y}^2)^{3/2}}\right) \end{pmatrix}$$

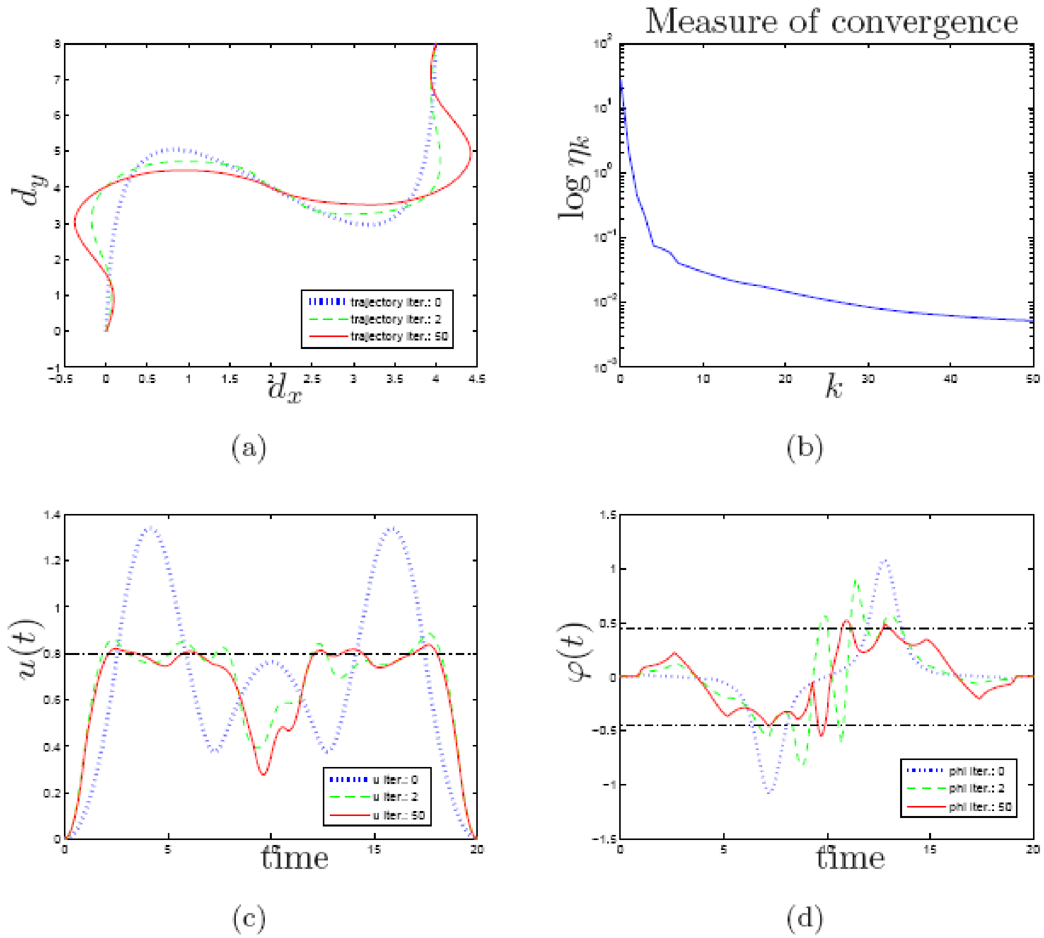


Figure 1: Initial reference trajectory, 2^{nd} and 50^{th} iteration. (a) Flat output $y = (d_x, d_y)$; (b) Measure of convergence η_k; (c) Input $u(t)$; and, (d) Input $\varphi(t)$.

$$\eta_k = \sum_{j=1}^{m} (P_j^{\text{ref,k}} - P_j^{\text{ref,k-1}})^{\text{T}} G_0^{\text{T}} G_0 (P_j^{\text{ref,k}} - P_j^{\text{ref,k-1}})$$