Nonlinear Model Predictive Path Tracking for Precision Guidance
Stavros Vougioukas\textsuperscript{1)}, Nick Sigrimis\textsuperscript{2)}, Konstantinos Arvanitis\textsuperscript{2)}, Yiannis Ampatzidis\textsuperscript{1)}

\textsuperscript{1)} Aristotle University of Thessaloniki, Department of Agricultural Engineering, 54124 Thessaloniki, Greece.
Tel. +2310998718, e-mail: bougis@agro.auth.gr , iampatzi@agro.auth.gr

\textsuperscript{2)} Agricultural University of Athens, Department of Agricultural Engineering, 11855 Athens, Greece.
Tel. +302105294036, email: n.sigrimis@computer.org, karvan@aua.gr

Introduction

In the context of precision farming, agricultural operations like planting, cultivation, tillage, fertilization, spraying, etc. require accurate position control of tractors and implements. Field coverage for such operations is typically defined by waypoint pairs, which are connected by parallel straight lines. A sequence of straight or curved lines defines a path and the goal of automatic path tracking is to minimize the average and maximum deviation between a tractor’s traveled path and the desired path. Various approaches have been proposed for path tracking, such as pure-pursuit (Amidi, 1990), dynamic path search control (Zhang and Qiu, 2004), and vector pursuit (Witt, et al., 2004). Modern auto-steering systems can provide accurate and repeatable path tracking of consistently straight or curved rows at an operator-set speed. It is expected that in the near future, precision farming operations will increasingly rely on more complex automatic steering and navigation capabilities of agricultural vehicles. Such capabilities include for example, sharp turns and reverse motions during headland turns, variable speed control, and eventually navigation among field obstacles. This problem is referred to as trajectory tracking and a desired trajectory point must be available to the tractor digital controller at every sample. The tracking error is defined at each controller sample as the difference between the desired and actual trajectory points. Various trajectory tracking controllers have been developed for industrial robots, with PID control being the most widely used technique. A well known theoretical result (Brockett, 1983) for under-actuated non holonomic systems, such as wheeled vehicles under the no-slip constraint, is that linear controllers cannot offer good tracking performance for complex maneuvers, sharp turns and reverse motions. Various control techniques have been proposed for trajectory tracking for car-like robots, such as time-varying LQR (Divelbiss and Wen, 1997), sliding mode control (Balluchi, et al., 1996), and iterative model predictive control (Wen and Jung, 2004). In this paper, a vehicle trajectory tracking controller is proposed, based on nonlinear
**Non Linear Model Predictive Control**

A brief description of the non linear model predictive control methodology is given next. Given an \( n \)-dimensional state space \( X \), an \( m \)-dimensional control space \( U \) and a discretization interval \( dt \), let a system’s discretized equation be of the form

\[
x_{k+1} = f(x_k, u_k), \quad k \geq 0
\]

(1)

where \( x_k \in X \), \( u_k \in U \) and \( f \) is the system’s state equation. The state and control vectors are subject to constraints of the form

\[
x_{\text{min}} \leq x_k \leq x_{\text{max}}, \quad u_{\text{min}} \leq u_k \leq u_{\text{max}}
\]

(2)

Let \( x_k^d, k = 0,1,...,N \) be a desired state trajectory, which has been computed off-line by some planning algorithm, and \( e_k = x_k^d - x_k^d \) be the error between the actual and the desired system trajectories. The basic idea behind nonlinear model predictive (NMP) control is to solve at every time step \( k \), a finite-horizon optimal control problem. Given the actual state \( x_k \), an optimal \( M \)-step-ahead control sequence \( u_{k,M}^* \) is computed, which minimizes a cost function \( J \).

\[
u_{k,M}^* = [u_{k}^*, u_{k+1}^*, \ldots, u_{k+M}^*] = \arg\min_{u_k, u_{k+1}, \ldots, u_{k+M}} J
\]

(3)

The cost function has the following form:

\[
J = \theta(e_{k+M+1}) + \sum_{j=k}^{\min(N,k+M)} \phi(x_j, u_j)
\]

(4)

where the tracking error of the last state of the finite horizon is penalized by a function \( \theta \) which typically has a quadratic form:

\[
\theta(e_{k+M+1}) = e_{k+M+1}^T Q e_{k+M+1}
\]

(5)

At each step, the state, tracking error and control effort along the optimizing horizon can be penalized also by a quadratic form:

\[
\phi(x_j, u_j) = x_j^T Q x_j + e_j^T Q e_j + u_j^T Q u_j
\]

(6)

In the NMPC framework, the control applied to the actual system at sample \( k \) is the first element of \( u_{k,M}^* \). At sample \( k+1 \) the system has moved to a new state \( x_{k+1} \) which differs from the predicted state due to disturbances and model errors. The optimization problem is solved...
again, until the system reaches its goal state $x_n^*$. Given that an optimal feasible open-loop control and trajectory for the problem exists, and that all the cost matrices $Q$ are positive-definite the stability of the closed-loop NMPC can be proved (Chen and Allgoewer, 1998).

**Path Tracking with NMPC**

In theory, given a tractor’s dynamic model, NMPC could be used as a tracking controller to control directly the steering angle and vehicle speed. Full dynamic models can be very complex, nonlinear, and include equations for modelling the tractor’s entire chassis (cabin, suspension, steering linkage, etc), the drive train (engine, gear box, shafts), and the tire-soil interaction. Such models involve many state variables and their NMPC optimization would require significant computation which may prohibit real-time operation. Another approach is to decouple the problem: use NMPC as a high-level tracking controller with a simple kinematical vehicle model, but incorporate dynamic models for steering and speed control. Such a simple kinematical model for a front-wheel steered tractor is given next.

\[
\begin{align*}
\dot{x} &= v \cos \theta, \\
\dot{y} &= v \sin \theta, \\
\dot{\theta} &= v \frac{\tan \phi}{L}
\end{align*}
\]  

(7)

The $x$ and $y$ coordinates give the position of the tractor’s rear-wheel axle midpoint and the tractor’s orientation is given by the angle $\theta$. The wheelbase length is $L$, the steering angle is $\phi$ and the vehicle speed is $v$. Of course, simple dynamic models, such as bicycle models (Wong, 1993) could be used for the NMPC, but still unknown model parameters related to tire-soil interaction need to be identified. At a lower-level, a steering controller model is used to create an “abstraction” of the steering system dynamics (Stombaugh, et al., 1998, Wu, et al., 1998); the same holds for speed controllers. In this work, the combination of the steering controller and the steering linkage is modeled as a first order system with time-lag $\tau_\phi$; an analogous abstraction is used for speed control.

\[
\begin{align*}
\dot{\phi} &= -\frac{1}{\tau_\phi} \phi + \frac{1}{\tau_\phi} u_\phi, \\
\dot{v} &= -\frac{1}{\tau_v} v + \frac{1}{\tau_v} u_v
\end{align*}
\]  

(8)

Hence, the NMPC tracker provides a desired steering angle $u_\phi$ to the steering controller and the actual wheels’ steering angle follows the command with first order dynamics. Similarly, velocity commands $u_v$ are issued by the NMPC to the speed controller. Of course, different models (e.g. 2$^{nd}$ order) could be used, as long as their parameters can be identified. From the NMPC point of view the vehicle is described by equations(7),(8) and the system state is
\( x = [x \ y \ \theta \ \phi \ \nu]^T \) and control \( u = [u_x \ u_y]^T \). The system equations must be discretized so that they can be expressed in the form of equation (1) and solved.

**Numerical NMPC Solution**

In the general case, NMPC optimization can only be solved numerically. In this paper the indirect approach was used, which uses gradient descent in order to minimize the problem’s Hamiltonian (Kirk, 1970). The Hamiltonian for NMPC optimization is

\[
H = \phi(x_k, u_k) + \lambda^T_{k+1} f(x_k, u_k) \tag{9}
\]

where \( \lambda_k \) is the costate sequence, which must satisfy the following equations

\[
\lambda_{k+M} = \frac{\partial \theta}{\partial x_{k+M}}, \quad \lambda_j = \frac{\partial H}{\partial x_j} + \frac{\partial f^T}{\partial x_j} \lambda_{j+1}, \quad j = k, ..., k + M - 1 \tag{10}
\]

The main idea behind the gradient descent algorithm is the following: if we start from a nominal solution \( u_{k,M}^\star \), which is close to the optimum, in order to minimize \( J \), a variation \( \delta u_{k,M}^\star \) of this control must be computed, such that the variation \( \delta J \) should be always negative:

\[
\delta J = \sum_{j=k}^{k+M} \left[ \frac{\partial H}{\partial u_j} \right]^T \delta u_j, \quad \text{where} \quad \frac{\partial H}{\partial u_j} = \frac{\partial \phi}{\partial u_j} + \frac{\partial f^T}{\partial u_j} \lambda_{j+1} \tag{11}
\]

This can be achieved by moving the control in the opposite direction of the Hamiltonian’s control gradient (steepest descent).

\[
u_{j+1}^\star = \nu_j^\star - K \frac{\partial H}{\partial u_j^\star}, \quad j = k, ..., k + M \tag{12}
\]

where the gain \( K \) is a sufficiently small positive number. The algorithm proceeds as follows: at each step \( k \), given an initial \( u_{k,M}^\star \), the gradient descent iteration index \( i \) is initialized to zero and equation (1) is used to compute the open-loop trajectory \( x_{k,M}^\star = [x_k^\star \ x_{k+1}^\star \ ... \ x_{k+M+1}^\star] \). Next, \( \lambda_j \) and \( \partial H / \partial u_j \) are computed using equations (10) and (11) respectively, and the updated controls \( u_{j+1}^\star \) are computed by equation(12). Finally, \( i \) is incremented and the iterations continue, until a convergence termination criterion is satisfied ( \( i > I_{\text{max}} \) or \( \delta J \geq 0 \)). Next, simulation results will be presented based on the described algorithm.

**Simulation Results and Conclusion**
The NMPC tracking was implemented in C++ and numerous simulations were performed. The tractor's wheelbase was 2 m, with \( \varphi_{\text{max}} = 60^\circ \) and \( v_{\text{max}} = 1 \text{ m/s} \). The integration step was set to 0.01s for increased accuracy, whereas the NMPC sampling period \( dt = 100 \text{ ms} \). All the elements of all cost matrices were set to zero. The diagonal elements of \( Q_e \) were set to \( q_e(1,1) = q_e(2,2) = 500 \) (distance error costs) and \( q_e(3,3) = 100 \) (orientation error cost). The velocity errors were penalized less, with \( q_e(5,5) = 50 \), because path accuracy was considered more important. The NMPC gradient descent algorithm was allowed to execute for 1000 iterations; larger iteration-limits did not improve the solutions. In a first experiment, the tracking algorithm was tested on a typical fish-tail headland manoeuvre. The nominal tractor velocity was equal to \( \pm 0.5 \text{ m/s} \) and the optimization horizon was set to \( M = 25 \). In Fig. 1 the convergence of the tracker can be seen for different starting states with large initial error. Each NMPC computation required 0.07s on a Pentium 2.6 GHz single-CPU system. Next the tracker’s performance was compared with that of a pure-pursuit type controller (PPC). The two paths are shown in Fig. 2. The PPC was tuned to provide critical damping and used a look-ahead distance of 1.25m (\( M = 25 \)). The average and maximum path tracking errors after the simulation were 4.62cm and 34.94cm respectively. The total area between the desired and executed paths was approximately 0.73m². The NMPC was initially executed with a horizon \( M = 25 \). However, this resulted in inferior performance compared to the PPC because the NMPC didn’t turn well in advance before the sharp turns. This is expected, since NMPC minimizes the tracking error over its entire horizon, whereas PPC turns \( M \) points in advance.

Next, the horizon was increased, and at \( M = 60 \) (5 meters), NMPC performed much better, with average and maximum tracking errors equal to 4.2cm and 19.78cm respectively. The total error-area was approximately 0.65m². The NMPC accomplished this by performing an
 anticipatory turning motion which increases the instantaneous tracking error, but leads to smaller total error. Such motions cannot be performed by PID-type controllers. The NMPC convergence time increased to 0.4s, but speed-problems could be overcome with dedicated hardware. From our simulations it was also clear that the choice of the cost matrices affects tracking quality because of orientation, velocity and position error trade-offs. Also, the optimization horizon $M$ is an important parameter, which regulates a trade-off between tracking quality (large $M$) and fast solution time (small $M$). If the NMPC solution time is longer than $dt$ (e.g. $\lambda dt$), then at the next $\lambda$ steps the optimal control computed at step $k$ can be used. This is equivalent to open-loop control and may lead to instability for large $\lambda$. As a conclusion, the NMPC tracker consistently converged to – and accurately followed - desired paths despite large initial errors and large path discontinuities. Hence, it seems to offer a promising approach for advanced precision guidance applications.

References