Non-linear Model Predictive Control of Constrained Mobile Robots

Harm van Essen and Henk Nijmeijer
Department of Mechanical Engineering, section Dynamics & Control
Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
Email: h.a.v.essen@tue.nl

Abstract
An application of Non-Linear Model Predictive Control (NLMPC) to the tracking and stabilisation of a kinematic model of a two wheel mobile robot with input and (non-holonomic) state constraints is investigated. Since for this system no smooth time-invariant stabilising feedback law exists and linear MPC is not feasible, the potential of non-linear predictive control techniques are investigated. The principles of NLMPC and its numerical implementation are discussed and results for several configurations are compared with existing time-varying and discontinuous feedbacks for the stabilisation of the mobile robot benchmark problem. The NLMPC controller outperforms the existing stabilising controllers with respect to convergence rates and steady state offset. The advantages of predictive control (constraint handling, optimal path planning etc.) are fully exploited. Still, the excessive computational demands make real-time implementation impossible at the moment.

1. Introduction
Model Predictive Control (MPC) is a control strategy that uses a model of the system that is to be controlled to obtain the control inputs by minimising an objective function. MPC is an essentially discrete time controller. At each sample interval, the model provides a prediction of process outputs over a future horizon. Based on this prediction, an objective function (usually reflecting the set-point deviations and actuator efforts) is optimised with regard to the future control inputs of the process. Although prediction and optimisation are computed over a future horizon, only the values of the inputs for the next sample are used and the same procedure is repeated at the next sample. This mechanism is known as a *moving* or *receding horizon* strategy. Due to the model based approach, the on-line optimisation can take into account constraints — limits on inputs or outputs, even ones that need not to be controlled — dead-time responses, and multivariable interaction. The inherent disadvantage of predictive control is the required computation time during each sample period. The computation time is a function of the length of the future horizons, the controller sample period, and the complexity of the models. Moreover, also the performance of MPC strongly depends on the quality of the model and of the selected values for tuning-parameters. An overview of MPC techniques can be found in [Camacho, 1999] or in [Garcia, 1989].

Mainly because of the attractive constraint handling capability and its intuitive "open-loop" approach, MPC with linear models is widely used in controlling relatively slow, constrained, processes as the large computational demands do not interfere the performance [Qin, 2000]. Although the computation time limits fast applications and usage of non-linear models so far, it is useful to question whether the use of non-linear models in the controller is worthwhile with respect to control performance at one hand and satisfying specific non-linear constraints on the other hand. In fact, in this context, not a strict distinction between linear and non-linear but between convex and non-convex problems should be made.

Linear MPC requires the solution of convex optimisation problems (usually quadratic programs, QP) for each sampling time, while non-linear MPC requires the consideration of a non-convex non-linear program, for instance sequential quadratic programming (SQP). Difficulties with NLMPC arise from both theoretical and computational perspectives. Due to the non-convexity, for instance, even nominal stability can no longer be guaranteed as a (global) feasible solution may not be found. Also the expense and reliability of solving the NLP on-line causes problems as it is hardly possible to guarantee convergence rates of the solution. An excellent overview of recent development on different aspects of non-linear MPC can be found in [Algöwer, 2000].

Since for most non-linear control problems a linear approximation gives satisfactory (local) results, here a non-linear mechanical benchmark system with non-holonomic constraints that cannot be solved by linear controllers has been selected to challenge the non-linear predictive controller. In recent years, non-linear control concepts for non-holonomic systems — like tracking and stabilisation of a two wheel mobile robot — attracted a lot of attention focussing on the development of suitable time-varying or non-smooth (discontinuous or switching) controllers. The NLMPC approach in this paper shows a number of interesting opportunities as opposed to these previously proposed controllers:

+ Constraints on both the inputs (saturation of actuators) and the states (or to be controlled outputs, or other constrained outputs) can be taken into account in a direct way. This means that they do not need to be secured by weighting factors.
+ Due to the open-loop, on-line, dynamic optimisation of predictive control, the resulting trajectory is optimal according to the optimisation criterion (including weightings). This also implies that no prescribed trajectory (or known solution) needs to be assumed for the controller design. In fact, the open-loop predictive controller functions as an optimal path-planning routine.
+ The intuitive "open-loop" approach of the predictive controller puts no demands on the structure of the non-linear model, and does not require (non-linear) state transformations to derive the control law. Therefore, the predictive controller is assumed to be more robust against modelling errors and disturbances compared to most of these analytically derived control laws.

The goal of this study is on the feasibility of NLMPC rather than contributing a new control scheme for non-holonomic systems. A goal, however, is to show the benefits and effectiveness of predictive control for complicated control tasks like the mobile robot. Obvious shortcomings and disadvantages of NLMPC, on the other hand, are the incredibly long computation time required by non-convex optimisation and the difficulties in guaranteeing stability and convergence.

In subsequent sections of this paper, first in section 2 the NLMPC problem and its implementation are introduced, including extensions like terminal or contractive state weightings or constraints. A new scheme including exponential set-point weighting over the prediction horizon is introduced. In section 3, the mobile robot control problem is introduced together with a brief review of previously proposed controllers. In section 4 results are presented emphasising on stabilisation with non-convex optimisation. For tracking control, i.e. control of a prescribed trajectory with persistently exiting inputs, results based on non-convex optimisation are compared with results based on successively linearised approximations. Finally, conclusions are drawn and new research is discussed.

2. Non-linear Model Predictive Control

First recall standard Linear Model Predictive Control, based on a linear discrete time model

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k+1) &= \mathbf{C} x(k+1) + \mathbf{D} u(k) \end{aligned} \qquad (1)$$

where $x \in \mathbf{R}^n$, $u \in \mathbf{R}^i$, $y \in \mathbf{R}^o$ and Φ, Γ, $\mathbf{C}$, and $\mathbf{D}$ are constant matrices of appropriate size.

Formulated in terms of the tracking error $e(t) = y(t) - r(t)$ with respect to a reference trajectory $r(t)$, and weighting *changes* in the input $\Delta u(k+1) = u(k+1) - u(k)$, the standard quadratic optimisation criterion is[1]:

$$\min_{\begin{bmatrix}\Delta u(k+1)\\ \ldots \\ \Delta u(k+m)\end{bmatrix}} \sum_{l=2}^{p} \left[\mathbf{Q}\left(y(k+l\,|\,k) - r(k+l)\right)\right]^2 + \sum_{l=1}^{m} \left[\mathbf{R}\left(\Delta u(k+l)\right)\right]^2 \qquad (2)$$

in which p is the number of samples of the prediction horizon, m is the number of samples of the control horizon, **Q** is the set-point weighting matrix, and **R** is the input weighting matrix. Figure 1 schematically presents the MPC scheme and introduces the adapted notation. Note that $y(k+l\,|\,k)$ stands for the (predicted) value of the output at sample $k+l$ made at or based on information that is available at sample k.

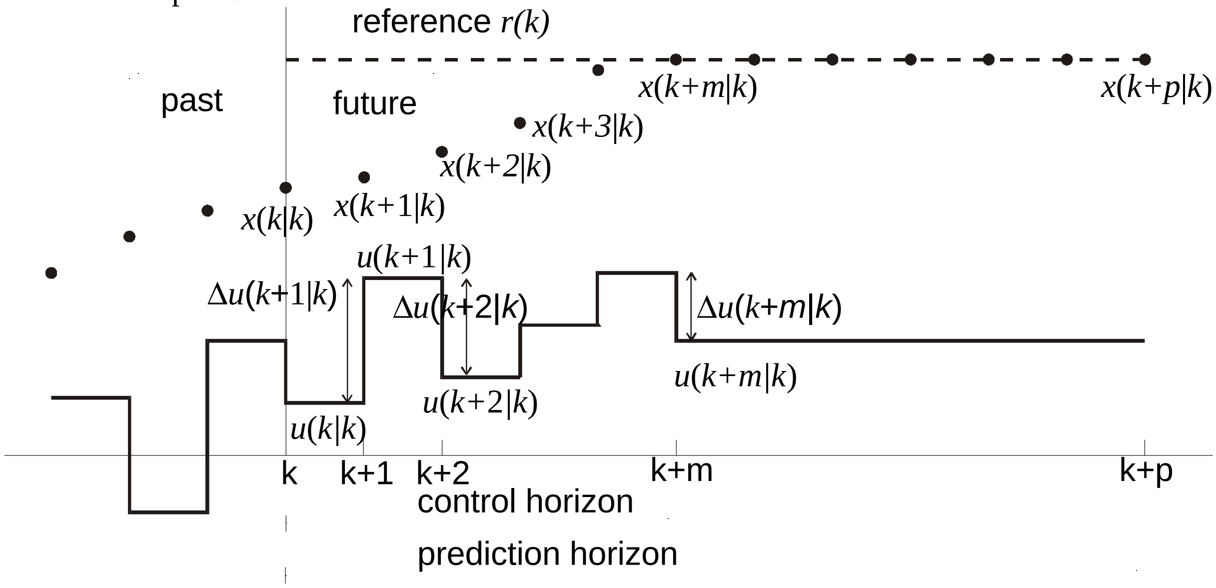

Figure 1 : MPC scheme indicating the sample numbering of states, inputs and input changes.

Criterion (2) equals a standard convex QP problem as

$$\min_{\Delta u} J = \tfrac{1}{2}(\Delta u)^T \left[\mathbf{Y}^T\mathbf{Q}\ \ \mathbf{Y} + \mathbf{R}\ \ \right](\Delta u) + \left(\mathbf{Y}^T\mathbf{Q}\, e_{\Delta u=0}\right)\Delta u \qquad (3)$$

in which by superposition the ($p\,o \times m\,i$) prediction matrix **Y** is defined by:

$$\begin{bmatrix} y(k+1) \\ \vdots \\ y(k+p) \end{bmatrix} = \underbrace{y_{\Delta u=0}}_{\substack{\text{future outputs}\\ \text{based on past}\\ \text{inputs}}} + \underbrace{\mathbf{Y}\begin{bmatrix} \Delta u(k+1) \\ \ldots \\ \Delta u(k+m) \end{bmatrix}}_{\substack{\text{future outputs based}\\ \text{on future inputs}}} \qquad (4)$$

For unconstrained problems an analytic solution to the QP problem is easily derived, and for constrained problems numerical solutions can be derived with efficient (standard) QP solvers. Also recall that the controller performance is highly influenced by the values of the above mentioned tuning parameters *p, m*, **Q** and **R** and of the controller sampling period of Δt.

---

1 Since only the value of Δu is weighted and not the reference value of the inputs, this standard criterion is only valid for constant set-point values or references trajectories with constant inputs.

Non-linear model based predictive control involves the solution of non-linear differential equations for model prediction and non-linear programming techniques for constrained optimisation. The NLMPC open loop optimisation problem is given by

$$\begin{aligned}\min_{u} \quad J = \Omega\big(x(k+p|k)\big) + \sum_{l=1}^{p} y(k+l|k)^T \mathbf{Q}\, y(k+l|k) \\ + \sum_{l=1}^{m} u(k+l|k)^T \mathbf{P}\, u(k+l|k) + \sum_{l=1}^{m} \Delta u(k+l|k)^T \mathbf{R}\, \Delta u(k+l|k)\end{aligned} \tag{5}$$

in which $\Omega(x(k+p|k))$ is a penalty (extra weighting) on the *terminal* state of the prediction horizon. Note that different weighting matrices for absolute inputs and changes in the inputs are applied. As superposition of prediction and optimisation is not applicable for non-linear models, **prediction and optimisation** have to be solved sequentially or simultaneously. In a sequential procedure, the model equations are simulated and the objective (and constraint) function is evaluated in an "inner-loop", while an optimisation code serves as an "outer-loop" [Sistu, 1993][Biegler, 2000]. This sequential approach is time consuming because the objective function may need to be evaluated over the prediction horizon many times. In a simultaneous procedure, the model equations are discretised (for instance by Euler's method) over the whole prediction horizon and included as non-linear equality constraints in a non-linear programming problem that minimises the objective function [Eaton, 1992]. This simultaneous approach results in very large optimisation problems, especially when the discretisation time steps need to be much smaller than the controller sample time to obtain accuracy. In spite of successful implementations and the advances made in developing efficient algorithms, [Biegler, 2000] the excessive computational requirements of both non-linear methods remain a serious obstacle for real-time implementation.

A computationally less demanding non-linear approach is linear approximation. The non-linear model equations are linearised around the current operating point to obtain locally a linear model. This can be done successively at each sample interval. Then the updated linear model is used for prediction and optimisation over the future horizon. It is also possible to repeat the linearisation within each sample of the prediction horizon. This extension, however, increases the computational demand substantially, since it incorporates a sequential (SQP) optimisation problem instead of a single QP problem over the prediction horizon. Another version of linearised MPC, introduced by [Garcia, 1984], uses a (successively) linearised model for optimisation combined with a non-linear model for prediction. This method combines attractive computational features with nonlinear performance, although the resulting closed-loop may be unstable.

In order to guarantee stability of the nominal NLMPC controller, additional (state) constraints have been introduced to the standard criterion (5). Two basic alternative formulations have been proposed [De Nicolao,2000]:

- NLMPC with a **terminal state equality or inequality constraint** [Chen, 1982],[Chen, 1998]. In this case the weighting on the terminal state is replaced by an equality constraint on the terminal state in the prediction horizon. Under the stringent assumption that for a given integer N a control sequence $u(k)$, $t<k<t+N-1$ can be found that drives the state from the initial condition towards the constraints in N steps, stability can be guaranteed.
- **Contractive NLMPC** [De Oliveira and Morari, 1996]. In this case either the terminal state of subsequent prediction horizons or the states over the current prediction horizon are forced to converge with a contraction factor $\alpha<1$. Stability can be guaranteed under the assumption that the optimisation problem has a feasible solution.

In this paper a new proposed algorithm with **exponential set-point weighting** attempts to improve the convergence without additional constraints. This means that the elements of the **Q** weighting matrix in (5) increase exponentially over the prediction horizon. The rationale is that additional weighting is easier solved than constraints. Also the strict assumptions on existence of solutions within a prediction horizon may be relaxed allowing some flexibility in the solution as it does not necessarily need to converge monotonically over the prediction horizon. Of course, when no (terminal state) constraints are included, (global) converging solutions cannot be guaranteed. Still, assuming feasible solutions may prove the stability of this alternative NLMPC scheme.

Difficulties with NLMPC arise not only from theoretical perspectives, but also from computational aspects. The expense and reliability of solving the SQP optimisation *on-line* causes problems. In practice, the sequential solution approach performs best, also because this is the most straightforward way implement NLMPC from the available model and constraint structure.

Several methods are available for (constrained) non-convex programs, of which Sequential Quadratic Programming (SQP) is the most advanced. A great deal of current research is on efficient solvers and problem definition tailored for NLMPC [Biegler, 2000]. In a numerical implementation standard SQP solvers can be chosen. The Matlab function FMINCON from the optimisation toolbox can be successfully applied. Also NAG routine E04UCF (available in Fortran, C and Matlab-MEX file) is designed to solve the non-linear programming problem. In general this problem is formulated as the minimisation of a smooth non-linear function $J(u)$, see criterion (5), subject to a set of constraints on the future input set u). The problem is assumed to be stated in the following form:

$$\underbrace{\min}_{u \in R^{m\times i}} F(u) \;\; s.t. \;\; b_l \le \begin{bmatrix} u \\ \mathbf{A}u \\ c(u) \end{bmatrix} \le b_u \qquad (6)$$

In which b_l and b_u are vectors of appropriate size determining the constraints. Obviously, actuator saturation bounds are included directly, while actuator rate constraints can be included straightforwardly as linear constraints in a matrix $\boldsymbol{A}$. Constraints on the outputs or on the states can only be included as (non-linear) constraints in a vector $\boldsymbol{c(u)}$**.** Such constraints complicate the optimisation procedure dramatically because each evaluation of the constraints involves the solution of the full system of non-linear model equations.

3. Mobile robot control problem

The tracking and stabilisation problem of a kinematic model of a two wheel mobile robot with input and (non-holonomic) state constraints is investigated. This is an interesting non-linear mechanical benchmark problem because it is well known [Brocket, 1983] that no smooth time-invariant stabilising feedback law exists.

Consider the mobile robot, schematically drawn in figure 2. The front wheel is free to rotate (castor) without any restrictions, and may therefore be considered as a gliding point or a rolling ball. Two independent forces can act on the rear wheels. Finally, a non-slipping condition on the movement holds. It is only possible to move in θ direction, sideway movements are restricted. This means that the **non-holonomic** constraint: $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$ holds.

Recall that this is by definition a non-holonomic constraint because it cannot be integrated, that is, written as a time derivative of a function of the generalised coordinates. More information on non-holonomic systems can be found in the review [Kolmanovski, 1995]. Since no side-way movement is possible therefore this is also considered as an **underactuated** control problem

A straightforward kinematic model with two independent inputs - the forward velocity u and the steering velocity w - of this benchmark problem can be given by:

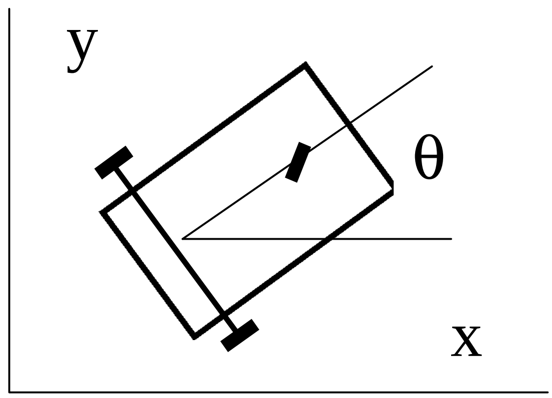

Figure 2 : Schematic view of the mobile robot in Cartesian co-ordinates.

$$\begin{cases} \dot{x}_1 = u\cos\theta \\ \dot{x}_2 = u\sin\theta \\ \dot{\theta} = w \end{cases} \qquad (7)$$

This model is a basic representative of a larger class of kinematic models representing non-holonomic systems like an equivalent mobile robot with four wheels, a truck with one or more trailers. These models can be easily extended to model with a dynamic input extension: i.e. torque as input instead of velocity.

First consider an intuitive approach to the mobile robot control problem. Realise that the non-linear model is fully controllable (imagine a car on empty parking place) but that the linearisation around an arbitrary stationary operating point $(x_0, y_0, \theta_0, u_0 = 0, w_0 = 0)$ is not controllable. The linearisation along a trajectory

$$\begin{cases} \dot{x}_{1r} = u_r(t)\cos\theta_r \\ \dot{x}_{2r} = u_r(t)\sin\theta_r \\ \dot{\theta}_r = w_r(t) \end{cases} \qquad (8)$$

however is controllable as long as the inputs are persistently exiting. This implies that stabilisation of the origin is not possible with linear approaches, while tracking of a reference trajectory (8) with persistently exiting inputs should be possible. In section 4, NLMPC results for stabilisation and tracking are presented. In the remaining of this section, two representative control schemes are discussed as introduction and reference for the NLMPC results. This paper does not pretend to give a complete overview of existing controllers.

Recent solutions to the problem of stabilising the origin of the mobile robot are found by defining suitable **time-varying feedback controllers —**the idea behind this is to avoid zero inputs by adding (for instance) a periodic signal with decreasing amplitude to the input— and **non-smooth (discontinuous) controllers —** the idea behind is to move towards the origin first as fast as possible, and then (locally) change the control law to stabilise the origin. Now consider two representative examples to illustrate these principles. Other results are obtained with the integrator backstepping design method [Jiang, 1997] and a cascaded design approach [Lefeber, 2000].

First consider an example of a time-varying feedback [Samson,1991]. For design and analysis of the controller, a non-linear, time-varying state transformation is introduced

$$\begin{cases} x = x_1\cos\theta + x_2\sin\theta \\ y = -x_1\sin\theta + x_2\cos\theta \\ z(t) = \theta(t) + f(y(t),t) \end{cases} \Rightarrow \begin{cases} \dot{x} = u + yw \\ \dot{y} = -xw \\ \dot{z} = w - \dfrac{\partial f}{\partial y}xw + \dfrac{\partial f}{\partial t} \end{cases} \qquad (9)$$

Choose for instance $f(y(t),t) = y(t)\sin(t)$. Then, by means of a Lyapunov method design a control law that stabilises the system can be derived:

$$\begin{cases} w = -z - \dfrac{\partial f}{\partial t} \\ u = -x + \dfrac{\partial f}{\partial y} zw \end{cases} \qquad \text{with} \qquad \begin{aligned} \frac{\partial f}{\partial t} &= y\cos(t) \\ \frac{\partial f}{\partial y} &= \sin(t) \end{aligned} \tag{10}$$

This results in asymptotic stabilisation, but a very slow convergence. See figure 3. The initial condition in original co-ordinates is [2, -1, 1]. Although tuning of the parameters and other selections of the time-varying function can significantly improve the performance, from this example be concluded that time varying controllers are not very intuitive, difficult to design and give slow convergence.

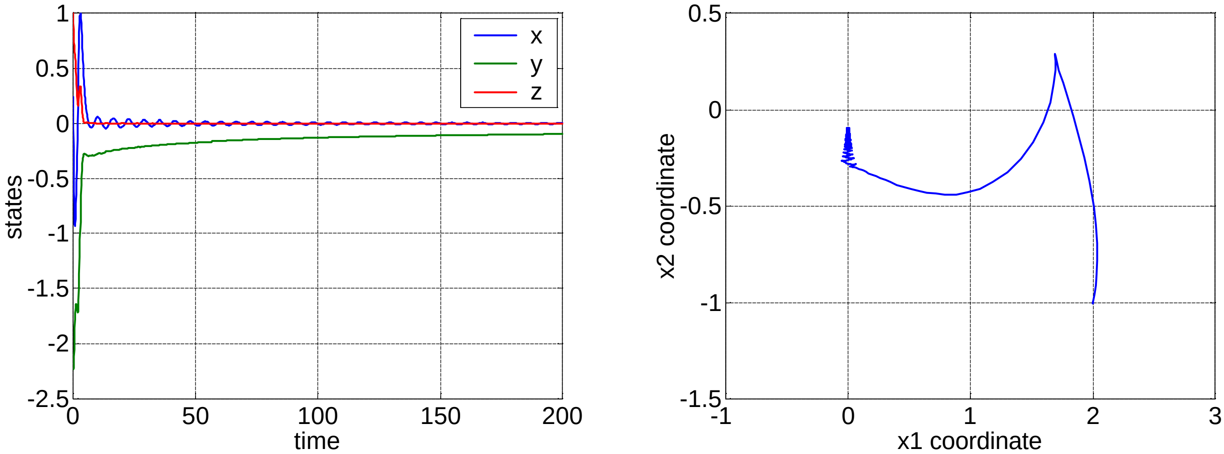

Figure 3: Time varying feedback law, results in both transformed co-ordinates x, y, z as well as original co-ordinates x_1 and x_2, controller parameters k_1=k_2=1.

A second example is a discontinuous controller [Astolfi,1996]:

$$\begin{bmatrix} u \\ w \end{bmatrix} = \begin{bmatrix} -k\left(\dfrac{x_1}{\cos\theta}\right) \\ p_2 x_2 + p_3 \dfrac{x_3}{x_1} \end{bmatrix} \tag{11}$$

With parameters chosen such that $\begin{bmatrix} k & -k \\ p_2 & p_3 \end{bmatrix}$ has eigenvalues with negative real part, this control law stabilises the original system (7) towards the origin. Results are presented in figure 4.

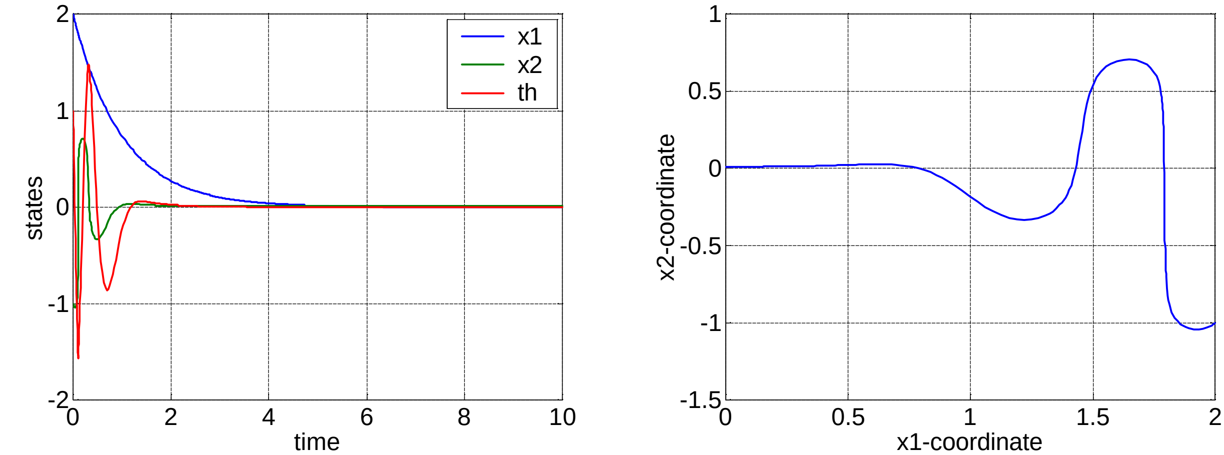

Figure 4: Discontinuous feedback law, controller parameters k=1, p2=25, p3=-5.

Note that the convergence rate towards the origin is determined by $\dot{x}_1 = -kx_1$ and that the controller is discontinuous at the surface $\{x \in \mathbf{R}^3 | x_1 = 0\}$. As the solution can not cross this surface, it always remains on one side, depending on the initial condition. Moreover, the convergence rate in x_2-direction is defined by $\dot{x}_2 = -kx_1 \tan\theta$, which is singular for $\theta = \pm\frac{\pi}{2}$ (causing unbounded input values) and zero for x_1=0 (causing steady state offsets). This also implies that due to numerical problems near the discontinuity, this controller is not able to stabilise the origin itself but only towards a neighbourhood of the origin.

A typical measure of convergence is the integrated error $\frac{1}{T}\int_0^T \left[x_1^2 + x_2^2 + \theta^2\right] dt$. Over the first 10 seconds, the time varying controller has an integrated error value of 1.94, while the discontinuous controller has a value of 0.28. Over longer time periods, however, the time-varying controller scores lower values as it approaches the origin asymptotically. For the discontinuous controller, numerical problems conceal the asymptotic behaviour.

In order to derive tracking controllers, often a state transformation towards the so-called error dynamics is used. This requires that a known, feasible, reference trajectory is available (8). The error dynamics are defined relative to this trajectory by means of a state transformation (12). It appears that the linearisation of this error dynamics along the reference trajectory is controllable as long as the inputs are persistently exiting.

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1r} - x_1 \\ x_{2r} - x_2 \\ \theta_r - \theta \end{bmatrix} \Rightarrow \begin{cases} \dot{x}_e = y_e w - u + u_r(t)\cos\theta_e \\ \dot{y}_e = -x_e w + u_r(t)\sin\theta_e \\ \dot{\theta}_e = w_r(t) - w \end{cases} \tag{12}$$

4. Results of NLMPC

This section presents simulation results obtained from an NLMPC implementation in Matlab on the kinematic model of the mobile robot benchmark problem. Emphasis is on the control objective stabilisation of the origin. In section 4.2 some results on tracking of a circle are briefly shown. As explained in the previous section, the tracking problem can be solved with a linear approximation as long as the inputs are persistently exiting. Therefore in section 4.2 results from non-linear optimisation are compared to results obtained with convex optimisation of linearised models.

In our results the matrix of weighting factors over the prediction horizon is an important tuning factor. We investigate the influence of a constant weighting of all the set-points over the prediction horizon and compare results with additional strong weighting of the terminal state and with exponentially increasing set-point weights. With respect to constraints we will investigate the influence of terminal state constraints on the performance and stability.

For reference, the other tuning parameters are fixed at arbitrary values (Dt = 0.2, p = m = 10 $\equiv 2\,\text{sec.}$, **R** = 0.1 for all inputs, **P** = 0 for all inputs). The initial conditions are for all simulation chosen equal at [x, y, θ, u, w] = [-0.5, 0.5, 1, 0.1, 0.1] and we use a model in the original Cartesian co-ordinates (7). The absolute values of the inputs are restricted. The forward velocity is bounded between -2 and 2 and the steering velocity is bounded between -3 and 3. There are no restrictions on the rate of change of the inputs.

Note that for all simulations a large computational effort is required: approximately 500 times real-time on a Pentium III-500 MHz!

4.1 NLMPC stabilisation results

First results are presented of a constant weighting of 1 for all set-points at all elements over the prediction horizon. Results in figure 5 clearly show a large steady state offset in at least one co-ordinate, in this case the y co-ordinate. Although, in contrast with an ordinary feedback controller, the predictive controller should be able to overcome this “dead-lock” situation by recognising in the prediction horizon the future gain that can be won, it does not. The only explanation is that the SQP solver is not able to generate persistently exiting inputs for the uncontrollable system in “dead-lock” situation.

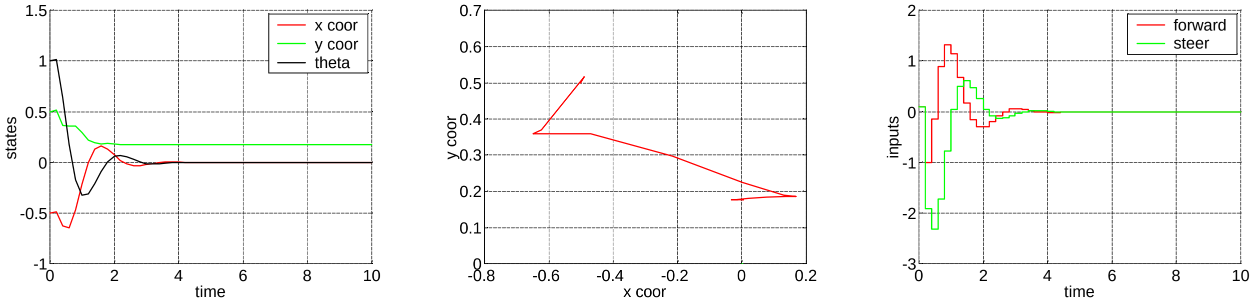

Figure 5 : Constant set point weighting [1 1 1] over the prediction horizon.

To eliminate the offset, additional weightings or constraints are required to force the system to the only allowable solution. One way to do so is to add a penalty on the terminal state in the prediction horizon according to $\Omega(x(k+p|k))$ in (5). Results in figure 6 are obtained with the same constant set point weighting over the prediction horizon and an additional terminal state weighting of 100 for each state. The results show less offset, but apparently, even (extreme) terminal weight only can not guarantee asymptotic convergence.

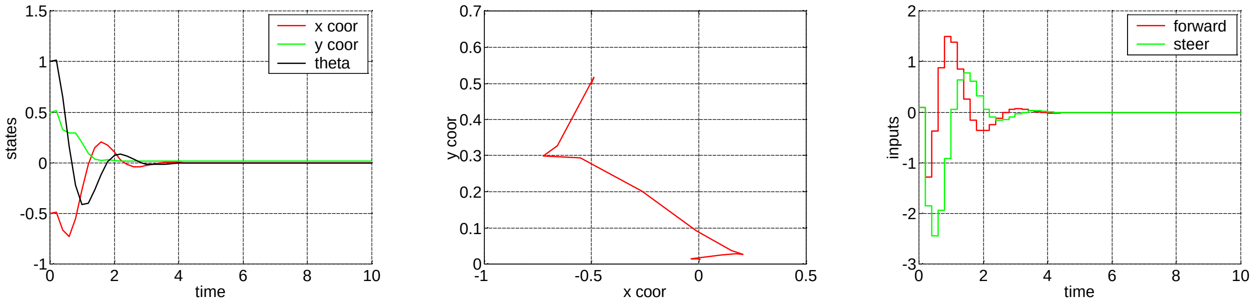

Figure 6 : Constant set-point weighting [1 1 1] over the prediction horizon and additional terminal state weighting [100 100 100].

Another way is to (exponentially) increase the weight on the set-point over the horizon. Results are very satisfactory: In figure 7, with a factor 2, no offset is seen and also the convergence rate is larger than in the previous cases. Note that now the absolute boundaries of the inputs are reached (and not violated), indicating maximum velocities. Indeed the zigzag response in figure 7 seems optimal with respect to the initial conditions and the input restrictions.

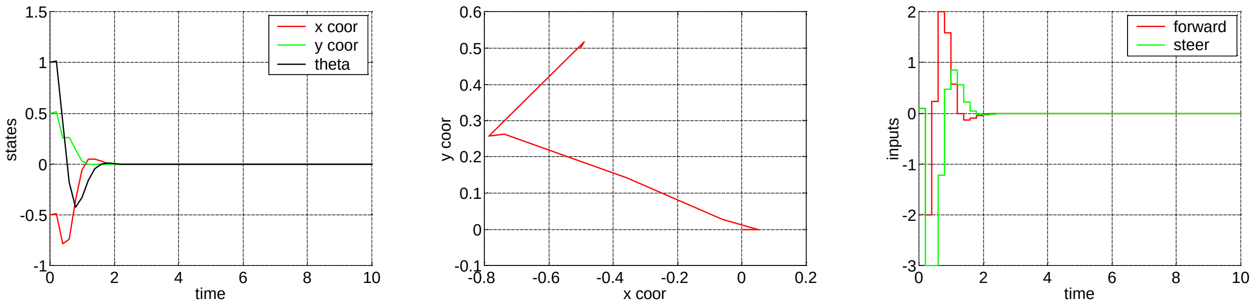

Figure 7 : Exponentially increasing set-point weighting over the prediction horizon (factor 2).

In the next two simulations, the previous results are compared to results with an additional constraint at the terminal state, *i.e.* $x(k+p \mid k)$ is forced to be zero. In figure 8 results of a constant weighting factor combined with a terminal state constraint are presented. Compare results with figure 5. Indeed a terminal constraint stabilises the system. The steady state response is offset free.

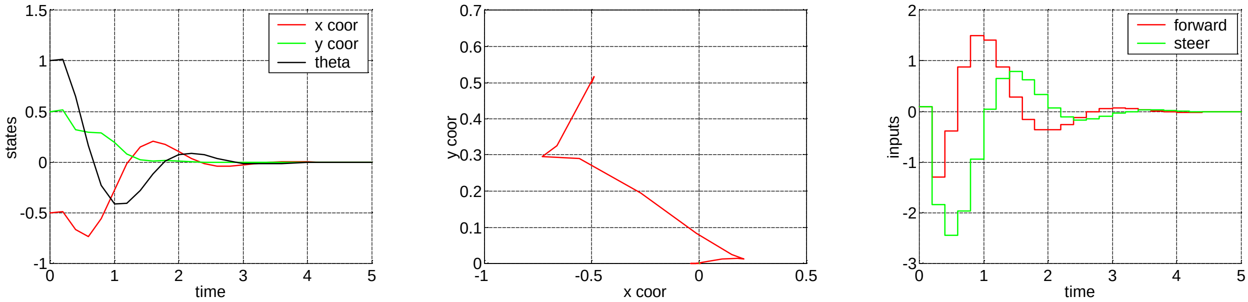

Figure 8 : Constant set-point weighting [1 1 1] over the prediction horizon combined with terminal state constraint at the origin [0 0 0].

Note, however, that the response is not very fast, certainly not faster than exponentially weighting only (see figure 7). Clearly, an optimal solution for which the terminal state $x(k+p \mid k)$ equals zero does not guarantee fast convergence because it does not mean that the solution also *stays* there: the corresponding inputs at the end of the prediction horizon are not (forced to be) zero. This means that additional set-point weighting still influences the performance significantly. In figure 9 the terminal state constraint is combined with exponentially increasing weighting over the prediction horizon, showing offset free very fast convergence although, again, response is not faster than exponential weighting only.

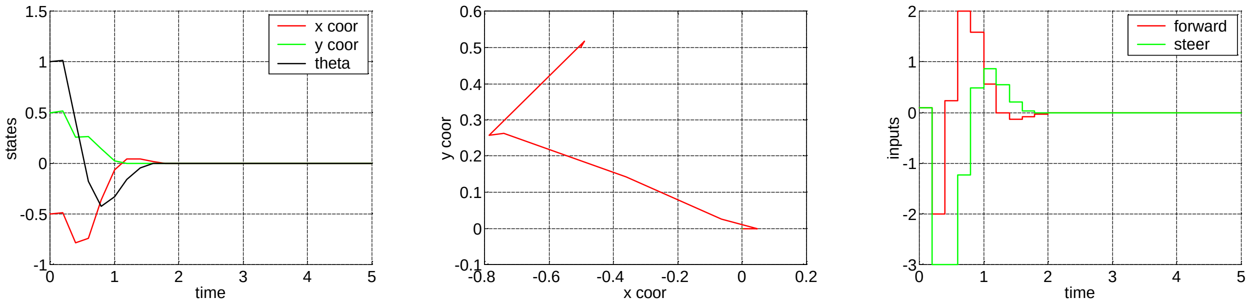

Figure 9 : Terminal state constraint at the origin [0 0 0], combined with exponentially increasing set-point weighting over the prediction horizon (factor 2).

Exponential weighting seems a good tuning parameter to obtain fast convergence. Indeed, additional weighting is easier solved than constraints and it provides faster convergence than terminal state constraints and more flexibility in solutions than contractive constraints over the prediction horizon.

4.2 NLMPC tracking results

In this section some results on the control objective **tracking** to a circle with large offset in the initial condition using the original model co-ordinates (7) are presented. The circle has a radius of 0.2. The starting point [x, y, θ] of the reference circle is [0, 0.2, 0]. One full circle is 10 seconds.

Because the reference inputs are persistently exiting (in fact they have a constant nonzero value), a successively linearised approach for optimisation (combined with a nonlinear model for prediction) is feasible for the tracking problem. The results in figure 10 however, clearly shows that non-linear optimisation (right hand side) gives faster convergence. For the non-linear optimisation problem with constant weighting of the set-point over the horizon, it takes about one full circle (10 seconds) before the reference path is tracked accurately. It can also be seen that the linear solution exhibits some peculiar behaviour. Indeed, the linear solution may get easily unstable or infeasible. For different settings of tuning factors the successively linearised problem cannot be solved because the optimisation is not feasible. We did not succeed, for instance, to solve exponential weighting or terminal state constraints in this linear approximation.

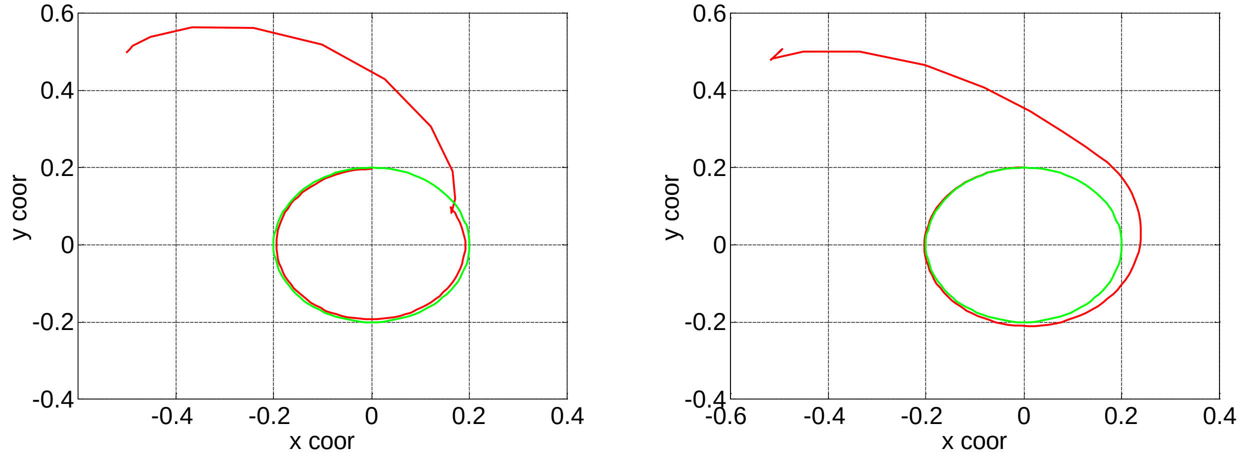

Figure 10 : Constant set-point weighting over the prediction horizon with Q = [1 1 1]. Successively linearised approximation (left) versus non-linear optimisation (right). Tuning parameters *p*= 20, *m* = 10, Δt=0.1. Initial point [x, y, θ, u, w]=[-0.5, 0.5, 1, 0.1, 0.1], start point reference circle [0, 0.2, 0], full circle is 10 seconds.

Figure 11 shows the significant influence of weighting and constraints for the non-linear approach. Again, exponential weighting gives the best performance in terms of fast convergence. Exponential weighting (left hand side) tracks the circle already after 1 second.

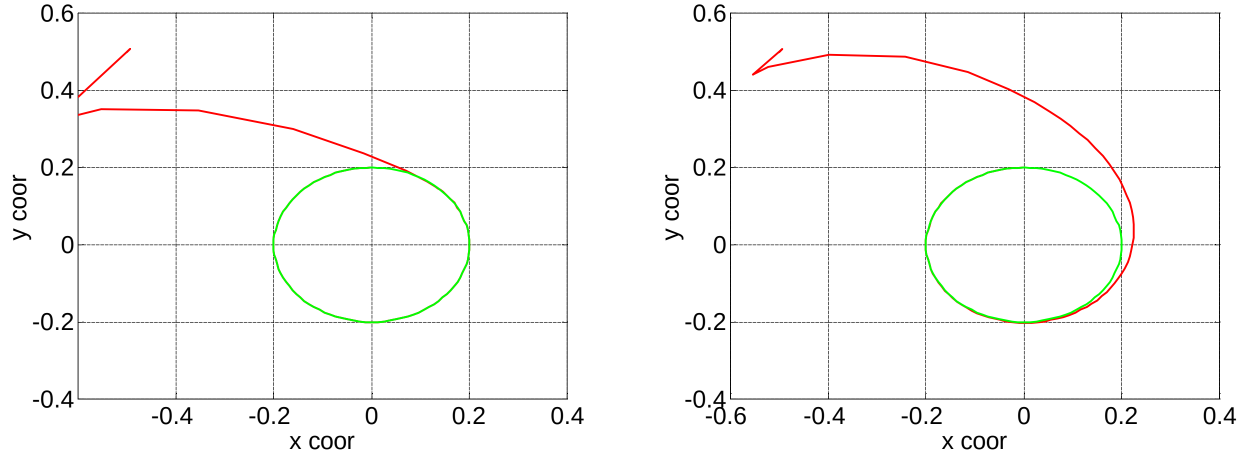

Figure 11 : Exponential set-point weighting (factor 2) (left) versus terminal state equality constraint (right). Otherwise identical initial position and tuning parameters as in previous figure.

5. Conclusions

Non-Linear Model Predictive Control (NLMPC) gives good tracking and stabilisation results for the kinematic model of a two wheel mobile robot benchmark problem. Results outperform representative time-variant and discontinuous feedback controllers for stabilisation with respect to convergence rate and steady state offset.

Indeed non-linear optimisation is required: linear optimisation can not solve the stabilisation problem because the linearisation around a stationary operating point is not controllable. Although the linearisation around the reference trajectory of a tracking problem is controllable under persistently exiting reference inputs, also in this case NLMPC outperforms the linear approximation with respect to convergence rate. Moreover, the linear approximation with successively linearised model for optimisation combined with a non-linear model for prediction is often not feasible.

The advantages of predictive control, like constraint handling of states and inputs, are also for NLMPC still available. Both for the stabilisation and the tracking problem the optimisation criterion of predictive control serves to find the optimal path to the set-point or reference trajectory, i.e. no off line path planner or prescribed solution is required. The major drawback of NLMPC is the excessive computation time that exceeds 500 times real-time for this basic three state model.

In our simulation results we find that tuning is decisive. Exponentially increasing set-point weighting is a new algorithm and successful in simulations. It assures convergence and therefore stability and gives faster convergence rates than (terminal) state constraints. Moreover, weightings are easier solved than state constraints and allow more flexible solutions.

6. Future Work

First, in developing a theoretical basis it is a challenge to determine the appropriate assumptions under which stability NLMPC with exponential set-point weights can be guaranteed. Second, in creating a practical basis, the development of efficient numerical implementation for real-time applications is important. Also work on (model based) moving horizon observers is important to estimate the required states for feedback. Finally, application on other non-linear (control) problems, thorough evaluation and comparison with other strategies will be conducted.

7. References

Allgöwer, F.; Zheng, A., editors, Nonlinear Model Predictive Control, Progress in Systems and Control Theory, Birkhäuser, Basel, 2000.

Astolfi, A.; Discontinuous control of nonholonomic systems, Systems and Control Letters, 27, p. 37-45, 1996.

Biegler, L.T.; Efficient Solution of Dynamic Optimisation and NMPC Problems, in Nonlinear Model Predictive Control, Frank Allgöwer and Alex Zheng, editors, Progress in Systems and Control Theory, Birkhäuser, Basel, 2000.

Brocket, R.W.; Asymptotic stability and feedback stabilisation, in Differential Geometric Control Theory, R.W. Brocket, R.S. Milman, and H.J. Susman, editors, Birkhäuser, Boston, 1983.

Chen, C.; Shaw, L.; On receding horizon feedback control, Automatica, Vol. 18, p. 349-352, 1982.

Chen, H.; Allgöwer, F.; A quasi-infinite horizon model predictive control scheme with guaranteed stability, Automatica, 1998.

Camacho, E.F.; Bordons, C.; Model Predictive Control, Springer, Londen, 1999.

De Nicolao, G.; Magni, L.; Scattolini, R.; Stability and robustness of nonlinear receding horizon control, in Nonlinear Model Predictive Control, Frank Allgöwer and Alex Zheng, editors, Progress in Systems and Control Theory, Birkhäuser, Basel, 2000.

De Oliveira, S.L.; Morari, M.; Contractive model predictive control for constrained nonlinear systems, Technical report, Automatic Control Laboratory, ETH Zurich, 1996. www.aut.ethz.ch

Eaton, John W.; Rawlings, James B.; Model predictive control of chemical processes, Chemical Engineering Science, Vol. 47, p. 705-720, 1992.

Garcia, Carlos E.; Quadratic dynamic matrix control of nonlinear processes, AIChE Annual Meeting, San Francisco, 1984.

Garcia, C.E.; Prett, D.M.; Morari, M.; Model Predictive Control: theory and practice - a survey, Automatica 25 (3) pp.335-348, 1989.

Jiang, Z.-P.; Nijmeijer, H.; Tracking control of mobile robots: a case study in backstepping, Automatica, Vol. 33, p. 1393-1399, 1997.

Kolmanovski, I.; McClamroch, N.H.; Developments in non-holonomic control problems, IEEE Control Systems Magazine, Vol. 16, p. 20-36, 1995.

Lefeber, E.; Tracking control of Nonlinear Mechanical Systems, PhD thesis, Twente University, Enschede, 2000.

Samson, C.; K.Ait-Abderramin; Feedback control of a nonholonomic wheeled cart in Cartesian space, Proceedings IEEE International Conference on Robotics and Automation, pp.1136- 1141, 1991.

Sistu, P.B.; Gopinath, R.S.; Bequette, B.W.; Computational issues in nonlinear predictive control, Computers and Chemical Engineering, Vol. 17, p. 361-366, 1993.

Qin, S.J.; An overview of Nonlinear Model Predictive Control Applications, in Nonlinear Model Predictive Control, Frank Allgöwer and Alex Zheng, editors, Progress in Systems and Control Theory, Birkhäuser, Basel, 2000.