Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Model Predictive Control with State Dependent Input Weight: an Application to Underwater Vehicles

Giancarlo Marafioti * Robert R. Bitmead ** Morten Hovd *

* *Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway (e-mail: [giancarlo.marafioti],[morten.hovd]@itk.ntnu.no).*
** *Department of Mechanical and Aerospace Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0411, USA (e-mail: rbitmead@ucsd.edu).*

Abstract:
Model predictive control (MPC) is an excellent approach for controlling systems with constraints. For nonlinear systems, nonlinear model predictive control (NMPC) is a natural solution, but it seems to be not suitable for system with relative fast dynamics. The main disadvantage is the time needed for solving the corresponding optimisation problem. Different approaches are used to simplify the optimisation problem and most of them yield an approximated optimal solution. It may happen that the best plant performances are obtained on the constraint borders. Thus, a controller that is able to handle and to work as close as possible to constraints without violating them, is desired. A faster computational power availability and improvements on the algorithms, make both the academic community and the industry to work intensively on the feasibility of using MPC on faster dynamics systems. In this work a model predictive control approach for controlling the depth of an underwater vehicle, with relatively fast nonlinear dynamics, is presented. Three different approaches are considered. For the internal model both a linear time invariant (LTI) and a linear time varying (LTV) model are implemented. A constant input weight is used for the LTI model, while for the LTV model also a state dependent input weight is utilised. The latter shows improvement on the control performance. An Extended Kalman Filter is used for state estimation.

1. INTRODUCTION

Model predictive control is a control technique that uses a mathematical model to predict the future response of the plant within a future time horizon. A specific performance index is defined by a cost function. The prediction is done every control interval, and together with the cost function, it is used to attempt an optimisation of the future plant behaviour. As result of the optimisation, a sequence of future manipulated variables is produced. The first element from the optimal sequence is chosen and applied as input to the plant, then the entire computation is repeated from the subsequent control interval. MPC is also recognised as a receding horizon control, because of the prediction window moving strategy.

MPC has been used and developed in industry for long time before the academic control community started to focus its attention on it. In Maciejowski (2002) it is observed that MPC is the only advanced control technique that has had a wide use in industrial process control. The main reason of its success is the ability to handle equipment and safety constraints. A survey of available commercial model predictive controllers, both linear and nonlinear, is provided in Qin and Badgwell (2003).

In linear MPC, a linear internal model is used to predict the system dynamics. However, the closed loop system can present a nonlinear dynamics due to the existence of constraints. Since many systems are described by nonlinear models nowadays, an active research is focusing on nonlinear model predictive control. The main reason is that a nonlinear internal model allows us to take into account the nonlinearities in the controller. Often the performance is improved by working as close as possible to the constraints. Then nonlinear model predictive control comes as a natural way of handling that. A comprehensive introduction on NMPC can be found in Findeisen and Allgöwer (2002). However, there are two complications relative to the NMPC: the stability issue and the computational burden.

The stability issue for linear MPC has been analysed and solved, for example by using a Fake Riccati Algebraic equation (Bitmead et al. (1990)), or using a terminal constraint (Mosca and Zhang (1992)), or a infinite horizon predictive control (Rossiter et al. (1996)). Concerning the NMPC stability, appreciable work is done in Michalska and Mayne (1993) where a terminal inequality constraints is added, such that the controller regulates the state into a feasible region. Furthermore, a local linear state feedback controller is designed to obtain the convergence, once the state is in that region, to an equilibrium point. In Magni and Sepulchre (1997) an alternative stability result is given by the use of a Fake Hamilton Jacobi Bellman equation. They showed that the stability margin of the receding horizon control is comparable to an optimal control law.

The computational burden is mainly due to the complexity of solving the constrained nonlinear optimisation problem on-line. Due to this the computational delay can be relevant. Moreover the non-convexity of the optimisation problem gives no guarantee of finding always a global optimal solution. The computational burden is particularly crucial in systems with relatively fast dynamics. A suitable method to solve the optimisation problem in a known time period is to formulate the problem as a convex optimisation problem by using a linear internal model. This was applied in Falcone et al. (2007) for the active steering of cars. Due to the fast dynamics involved, they used a linear time varying MPC based on the on-line successive linearisation of the nonlinear vehicle model. Another interesting approach is the gain-scheduling MPC approach discussed in Chisci et al. (2003), where a considerable part of the computation is executed off-line. In Sutton and Bitmead (1998) a model predictive control of an underwater vehicle is described and compared to LQG control.

In this work a model predictive control approach for controlling the nonlinear dynamics of an underwater vehicle, relatively fast, is presented where a linear internal model is used to reduce the computational load. It is shown how the introduction of a state dependent weight in the cost function leads to improved control performance with respect to the use of a constant input weight. The state dependent weight idea in the cost function, resembles the use of a varying controller gain adopted in the gain scheduling techniques (see Rugh and Shamma (2000)). By assuming a constant surge vehicle speed, the control goal is to maintain a constant distance from the ocean bottom, using the rudder angle as control variable. The distance from the ocean bottom is assumed to be known (for example measured with a Doppler velocity log (DVL) in its usual bottom-looking configuration). An Extended Kalman Filter is used to estimate the remaining state of the system.

In the next section the vehicle model equations are stated. In the third section a general formulation for the model predictive control is described. In the fourth section a state dependent nonlinear weight is defined, and its benefits are explained. In the fifth section the simulation results are presented and then conclusions are given in the sixth section.

2. TWO DIMENSIONAL UNDERWATER VEHICLE MODEL

Figure 1 shows a sketch of the vehicle with the reference frames. The vehicle dynamics is given in the body-fixed frame defined by Cxz. The surge and heave speed are defined by u and w, respectively. The pitch angle rate is denoted q. In the earth-fixed reference frame OXZ we describe the motion of the vehicle as shown in (2). The attitude of the vehicle is defined by θ_c.

The vehicle dynamics is described, according to Sutton and Bitmead (1998), as

$$M_I \begin{pmatrix} \dot{u}(t) \\ \dot{w}(t) \\ \dot{q}(t) \end{pmatrix} = mq(t) \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u(t) \\ w(t) \\ q(t) \end{pmatrix} + u(t) D_h \begin{pmatrix} u(t) \\ w(t) \\ q(t) \end{pmatrix} + \Gamma_g(t) + U_{cw}(t) + U_{cp}(t) \tag{1}$$

$$\begin{pmatrix} \dot{x}_c(t) \\ \dot{z}_c(t) \\ \dot{\theta}_c(t) \end{pmatrix} = \begin{pmatrix} \cos(\theta_c(t)) & \sin(\theta_c(t)) & 0 \\ -\sin(\theta_c(t)) & \cos(\theta_c(t)) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u(t) \\ w(t) \\ q(t) \end{pmatrix} \tag{2}$$

where M_I is the inertia matrix including the hydrodynamic added mass, m is the vehicle mass, D_h is the damping matrix, and the buoyancy term $\Gamma_g(t)$ is zero because the vehicle is assumed to be neutrally buoyant. $U_{cw}(t)$ and $U_{cp}(t)$ are respectively the forces and the moments generated by the rudder and propeller. Their expressions are given by equations (3)-(12).

$$V^2 \triangleq u^2 + w^2 \tag{3}$$

$$\epsilon \triangleq \sin^{-1}\left(\frac{w}{V}\right) \tag{4}$$

$$J_p \triangleq \frac{u}{|\nu D_p|} \tag{5}$$

where ϵ is the angle between the Cx axis and the velocity vector $\mathbf{V}$. J_p is the propeller advancement coefficient, ν is the propeller shaft rotational speed, and D_p is the propeller diameter.

$$C_{xsw} \triangleq -\left(C_{xow} + c_w \frac{C_{zow}^2(\epsilon+\beta)^2}{2\pi b_w}\right) \tag{6}$$

$$C_{zsw} \triangleq -\left(C_{zow}(\epsilon+\beta) + 2.1(\epsilon+\beta)^2\right) \tag{7}$$

$$U_{cw} = \left(U_{cw_{11}} \quad U_{cw_{21}} \quad U_{cw_{31}}\right)^T \tag{8}$$

$$U_{cw_{11}} = 0.5\rho S_w V^2 (C_{zsw}\sin(\beta) + C_{xsw}\cos(\beta)) \tag{9}$$

$$U_{cw_{21}} = 0.5\rho S_w V^2 (C_{zsw}\cos(\beta) - C_{xsw}\sin(\beta)) \tag{10}$$

$$U_{cw_{31}} = -U_{cw_{21}}(0.2c_w\cos(\beta) + d_{aw}) - U_{cw_{11}}(0.2c_w\sin(\beta)) \tag{11}$$

$$U_{cp} = \begin{pmatrix} \rho|\nu|\nu D_p^4(C_{t0p} + C_{t1p}J_p + C_{t2p}J_p^2 + C_{t3p}J_p^3) \\ -\rho|\nu|D_p^3 w C_n \\ \rho|\nu|D_p^3 w C_n D_{ap} \end{pmatrix}. \tag{12}$$

where $S_w = b_w c_w$ is the rudder surface, and ρ is the sea water density. C_{xow}, C_{zow}, d_{aw}, b_w, c_w, are the rudder characteristics, and C_{t0p}, C_{t1p}, C_{t2p}, C_{t3p}, C_n, D_{ap}, are the propeller characteristics (more details can be found in Santos and Bitmead (1995)).

The vehicle has two inputs, they are respectively the rudder deflection β, and the propeller rotation frequency ν, and they can be used for control purpose.
The ocean bottom position is modelled such that it will be treated as a disturbance to be rejected by the controller. Its model is described by the state x_f which is the rate of change of the absolute angle of the ocean bottom defined by $\theta^*(t)$ at $x^*(t)$. The angular velocity of the sea floor $\dot{\theta}^*(t)$ is modelled as the negative output of a first order filtered white noise process driven by $\xi(t)$ (equations 13-14):

$$\dot{x}_f(t) = A_f x_f(t) + B_f \xi(t) \tag{13}$$

$$f(t) = C_f x_f(t). \tag{14}$$

The relative angle θ between the ocean bottom and the vehicle is given by

$$\theta(t) = \theta_c(t) - \theta^*(t) \tag{15}$$

which gives

$$\begin{aligned} \dot{\theta}(t) &= \dot{\theta}_c(t) - \dot{\theta}^*(t) \\ &= q(t) + C_f x_f(t). \end{aligned} \tag{16}$$

where $\dot{\theta}^*(t) = -f(t) = -C_f x_f(t)$.
The relative distance between the ocean bottom and the vehicle centre is given by $\kappa(t)$ and its rate of change is computed with

$$\dot{\kappa}(t) = u(t)\sin(\theta(t)) - w(t)\cos(\theta(t)). \tag{17}$$

We assume we can measure the distance between the vehicle centre and the ocean bottom, and the measurement is affected by a white Gaussian noise η:

$$y(t) = \kappa(t) + \eta(t). \tag{18}$$

The system (1-17) can be written in compact form as

$$\dot{x}(t) = f\left(x(t), \beta(t), \xi(t)\right) \tag{19}$$

where $x(t) = (u, w, q, \theta, \kappa, x_f)^T$ is the state vector, and where the measurements are given by (18). The vehicle parameters can be found furthermore in Santos (1995). The system (18)-(19) has a non-minimum phase behaviour, in the sense that a positive input step change on β makes the vehicle to accelerate in the opposite direction for converging eventually to the steady state value of the response. Furthermore, an inability to accelerate in an arbitrary direction, due to the limited range of control value reachable U_{cw}, makes the system nonholonomic. These considerations make the control of the vehicle interesting and suitable for investigating the model predictive control features.
In this work a discrete time framework is used, then a discrete time model is needed, and it can be easily obtained by using the Euler approximation

$$\dot{x}(t) \simeq \frac{x_{k+1} - x_k}{h} \tag{20}$$

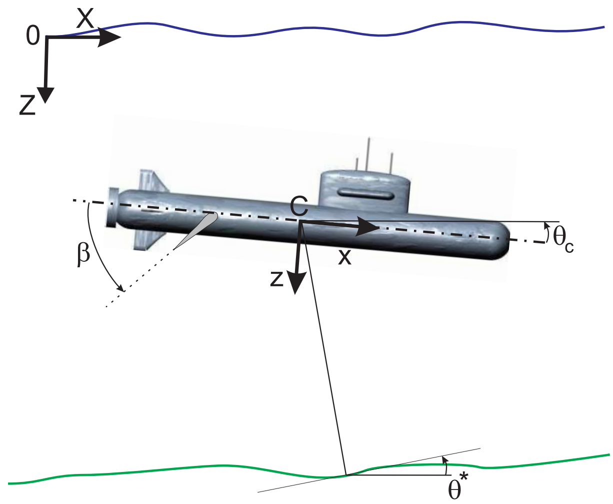

Fig. 1. Vehicle reference frames

where h is the sampling time. The correspondent discrete time model is then written as

$$\begin{aligned} x_{k+1} &= f(x_k, \beta_k, \xi_k, h) \\ y_k &= g(x_k, h). \end{aligned} \tag{21}$$

3. MODEL BASED PREDICTIVE CONTROL

3.1 Formulation

For a generic discrete nonlinear system

$$\begin{aligned} x_{k+1} &= f(x_k, \beta_k) + \omega_k \\ y_k &= g(x_k, \beta_{k-1}) + v_k \end{aligned} \tag{22}$$

where k is the sampling time index, x is the state vector, β is the input vector, y is the measurement vector, ω and v are Gaussian white noise vectors. The cost function to minimise is

$$\begin{aligned} \min_{\bar{\beta}} f(x, \beta) = \sum_{i=0}^{Np-1} & \{(x_i - x_{i_{ref}})Q(x_i - x_{i_{ref}}) + \\ & (\beta_i - \beta_{i_{ref}})P(\beta_i - \beta_{i_{ref}})\} + \\ & (x_{N_p} - x_{N_{p_{ref}}})S(x_{N_p} - x_{N_{p_{ref}}}) \end{aligned} \tag{23}$$

subjected to the following constraints

$$\begin{aligned} & x(t=0) = x_0 \\ & \beta_{L_i} \le \beta_i \le \beta_{U_i}, \quad 0 \le i \le N_p \\ & x_{L_i} \le x_i \le x_{U_i}, \quad 0 \le i \le N_p \end{aligned} \tag{24}$$

where $\bar{\beta} = \{\beta_0, \beta_1, ..., \beta_{Np}\}$, N_p is the prediction horizon length, and the subscript ref indicates the reference vectors. In the constraint inequalities the subscript L and U indicate the lower and upper bound, respectively. The weights Q and S are positive semidefinite matrices, and P is a positive definite matrix.

3.2 Internal models used for prediction

We want to use a linear internal model such that a convex optimisation problem is obtained. Two different approximations of (22) are employed. The first one is a linear time invariant (LTI) model

$$\begin{aligned} x_{i+1} &= Ax_i + B\beta_i \\ y_i &= Cx_i \end{aligned} \tag{25}$$

where A, B, and C are the matrices obtained by linearising (22) at the origin.
The second one is a linear time varying (LTV) model

$$\begin{aligned} x_{i+1} &= A_i x_i + B_i\beta_i \\ y_i &= C_i x_i \end{aligned} \tag{26}$$

where A_i, B_i, and C_i are computed by:

$$\begin{aligned} A_i &= \left.\frac{\partial f(x,\beta)}{\partial x}\right|_{x=x_i^*,\beta=\beta_i^*} \\ B_i &= \left.\frac{\partial f(x,\beta)}{\partial \beta}\right|_{x=x_i^*,\beta=\beta_i^*} \\ C_i &= \left.\frac{\partial g(x)}{\partial x}\right|_{x=x_i^*,\beta=\beta_i^*} \end{aligned} \tag{27}$$

where $i = \{1, 2, ..., N_p\}$, β^* is the tail of the optimal solution from the previous time step, and x^* is the state obtained when β^* is applied to (22). The complete algorithm is described in the next section.

4. NONLINEAR STATE DEPENDENT COST FUNCTION WEIGHT

The model predictive control cost function is chosen as in Sutton and Bitmead (1998):

$$J = \sum_{i=0}^{N_p} y_i^2 + P_c\beta_i^2 + R_c\theta_i^2 \tag{28}$$

where P_c is the input weight, and R_c is a penalty on the angle θ in order to avoid that the vehicle heads backward. The cost function (28) can be written in the form of (23) where
$x_{i_{ref}} = (0, 0, 0, 0, 0)^T$, $\beta_{i_{ref}} = 0$, $x_i = (w_i, q_i, \theta_i, \kappa_i, x_{f_i})^T$,
$Q = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & R_c & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$, $P = P_c$, and $S = Q$.
As it is shown in the next section, using the model (25) and the cost function (28) the controller is able to regulate the output, but only if the vehicle starts from a depth relatively close to the reference, Figs. (2-3).
To increase the region of attraction a nonlinear input weight P_c is defined as follow:

$$P_c = P_c(k) = a\kappa_k^2 + b \tag{29}$$

where κ is the relative depth at time step k, a and b are positive constant design parameters. Thus, for every time step a new input weight is calculated. An estimate $\hat{\kappa}$ of the relative depth is obtained using an Extended Kalman Filter. The input weight is then computed by

$$P_c(k) = a\hat{\kappa}_k^2 + b. \tag{30}$$

4.1 Algorithm

The MPC algorithm is applied to the time discrete nonlinear system (21). For a general time step k we:

(1) use β_k and y_k to estimate the state $\hat{x}_k$;
(2) use the tail of the optimal solution at the previous step $\beta^*|_{k-1} = \left\{\beta_2^*, \beta_3^*, ..., \beta_{N_p}^*, 0\right\}$ to compute the future state x^*, within the prediction horizon N_p;
(3) compute the Jacobians (27) about the predicted x^* and β^*;
(4) calculate the state dependent nonlinear weight (30);
(5) find the optimal solution of the corresponding QP optimisation problem;
(6) from the optimal solution $\beta^*|_k = \left\{\beta_1^*, \beta_2^*, ..., \beta_{N_p}^*\right\}$ apply β_1^* to the vehicle, increment k and go to the algorithm step (1).

5. SIMULATION RESULTS

The MPC controller has been simulated in the Matlab environment. Some relevant simulation parameters are: sampling period $h = 0.2s$, input constraints $|\beta| \leqslant 0.5rad$, cost function weights $P_c = 150$ and $Rc = 10$ when constant weights are used, $a = 0.2$, and $b = 5$ when the nonlinear input weight (30) is applied. The QP problem is solved on-line with the standard Matlab QP solver quadprog.
The control goal was to make the vehicle follow the ocean bottom with a 50m offset. Figure 2 shows the result of a simulation carried out using the LTI model and the cost function (28) with the constant input weight P_c. The vehicle was able to achieve its task, only when starting from of maximum distance from the bottom of about 70. As expected this result is due to the linear model obtained by linearising the system around the origin. In fact when the vehicle starts too far away from the reference, the linearised model does not represent accurately the real system. This is shown in (3) where the vehicle, starting from 50m above the reference, fails to converge to the assigned depth.
By employing the constant input weight and the LTV model (26) instead of the LTI one, the region of attraction becomes bigger. Figures 4 and 5 show the controller performance when the vehicle starts at 20 and 50 meters above the reference, respectively. In both cases the tracking error converges to zero, but in the second case the control input presents large oscillations that could damage the actuator system.
Figures 6 and 7 are obtained employing the LTV model with the state dependent input weight (30). The results are improved with respect to the use of the constant input weight. The region of attraction is larger and the control input does not present oscillations.

6. CONCLUSIONS

In this work a model predictive control scheme for the depth control of an underwater vehicle was implemented. Three different approaches were considered. For the internal model both a linear time invariant and a linear time

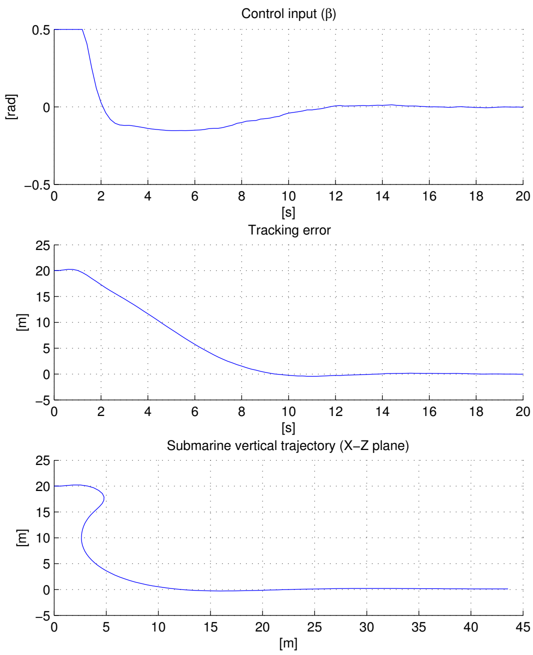

Fig. 2. Controlled input β, tracking error, and trajectory in the vertical plane (with respect to the 50m vertical offset) using the LTI model (25) and constant cost function weights (initial tracking error 20m).

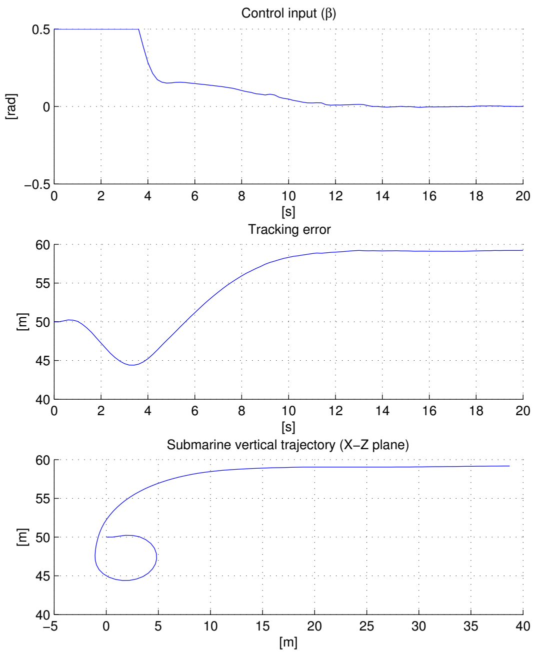

Fig. 3. Controlled input β, tracking error, and trajectory in the vertical plane (with respect to the 50m vertical offset) using the LTI model (25) and constant cost function weights (initial tracking error 50m).

varying model were implemented. For the LTI model, a constant input weight was used while for the LTV model also a state dependent input weight was utilised. The latter showed improved control performance in terms of region of attraction and convergence time.
When the vehicle is moving along the vertical, that is for $\theta = -\pi/2$, $\omega = 0$, and $q = 0$, the linearised system is unobservable. Future work is needed to analyse the possibility of using the MPC prediction feature to deal with this issue.

REFERENCES

R.R. Bitmead, M. Gevers, and V. Wertz. *Adaptive optmimal control, the thinking man's GPC*. Prentice Hall, 1990.

L. Chisci, P. Falugi, and G. Zappa. Gain-scheduling mpc of nonlinear systems. *International journal of robust and nonlinear control*, 13:295–308, 2003.

P. Falcone, F. Borrelli, H.E. Tseng, J. Asgari, and D. Hrovat. Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation. *International journal of robust and nonlinear control*, (In press), 2007.

R. Findeisen and F. Allgöwer. An introduction to nonlinear model predictive control. In *21st Benelux Meeting on Systems and Control*, 2002.

J.M. Maciejowski. *Predictive control with constraints*. Prentice Hall, 2002.

L. Magni and R. Sepulchre. Stability margins of nonlinear receding horizon control via inverse optimality. *Systems and Control Letters*, 32(4):241–245, 1997.

H. Michalska and D.Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE transactions on automatic control*, 38(11):1623–1633, 1993.

E. Mosca and J. Zhang. Stable redesign of predictive control. *Automatica*, 28(6):1229–1233, 1992.

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

J.A. Rossiter, J.R. Gossner, and B. Kouvaritakis. Infinite horizon stable predictive control. *IEEE transactions on automatic control*, 41(10):1522–1527, 1996.

W.J. Rugh and J.S. Shamma. Research on gain scheduling. *Automatica*, 36:1401–1425, 2000.

A. Santos. PhD thesis, L'École Nationale Supérieure des Mines de Paris, 1995.

A. Santos and R.R. Bitmead. Nonlinear control for an autonomous underwater vehicle (auv) preserving linear design capabilities. In *34th IEEE CDC. New Orleans, Louisiana: USA*, 1995.

G.J. Sutton and R.R. Bitmead. Experiences with model predictive control applied to a nonlinear constrained submarine. In *37th IEEE CDC. Tampa, Florida: USA*, 1998.

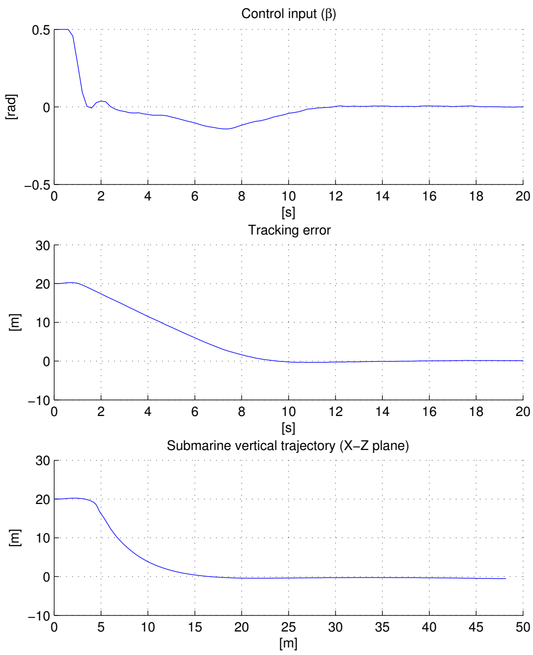

Fig. 4. Controlled input β, tracking error, and trajectory in the vertical plane (with respect to the 50m vertical offset) using the LTV model (26) and constant cost function weights (initial tracking error 20m).

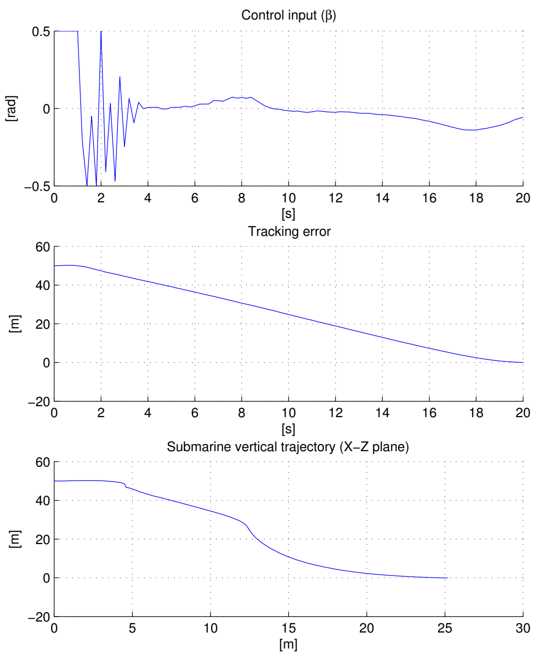

Fig. 5. Controlled input β, tracking error, and trajectory in the vertical plane (with respect to the 50m vertical offset) using the LTV model (26) and constant cost function weights (initial tracking error 50m).

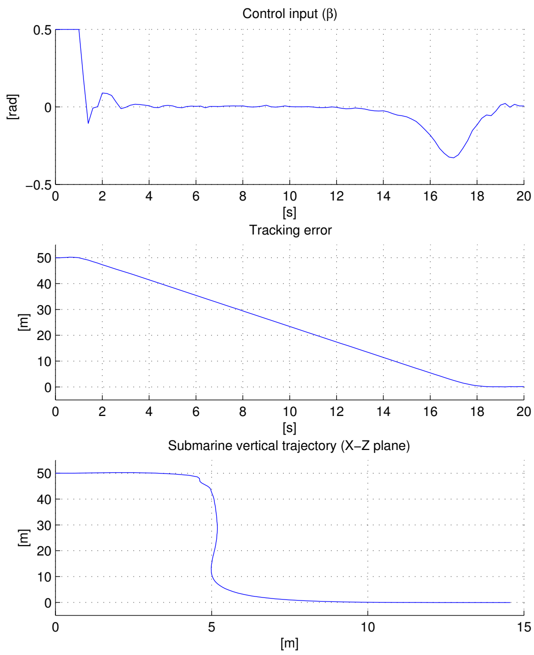

Fig. 6. Controlled input β, tracking error, and trajectory in the vertical plane (with respect to the 50m vertical offset) using the LTV model (26) and state dependent input weight (initial tracking error 50m).

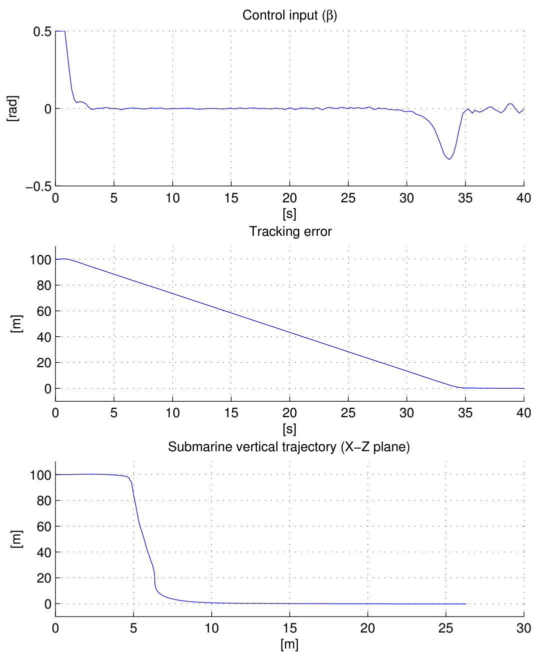

Fig. 7. Controlled input β, tracking error, and trajectory in the vertical plane (with respect to the 50m vertical offset) using the LTV model (26) and state dependent input weight (initial tracking error 100m).