



This is a repository copy of *Dual-mode model predictive control of an omnidirectional wheeled inverted pendulum*.

White Rose Research Online URL for this paper:
<http://eprints.whiterose.ac.uk/151418/>

Version: Accepted Version

Article:

Watson, M.T. orcid.org/0000-0001-8443-4372, Gladwin, D.T., Prescott, T.J. et al. (1 more author) (2019) Dual-mode model predictive control of an omnidirectional wheeled inverted pendulum. *IEEE/ASME Transactions on Mechatronics*, 24 (6). pp. 2964-2975. ISSN 1083-4435

<https://doi.org/10.1109/tmech.2019.2943708>

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works. Reproduced in accordance with the publisher's self-archiving policy.

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Dual-mode Model Predictive Control of an Omnidirectional Wheeled Inverted Pendulum

Matthew T. Watson¹, Daniel T. Gladwin², Tony J. Prescott³, and Sebastian O. Conran⁴

Abstract—This article describes the position and heading control of a novel form of omnidirectional wheeled inverted pendulum platform known as a Collinear Mecanum Drive. This concept uses four collinear Mecanum wheels to balance in a similar manner to a typical two-wheeled inverted pendulum, whilst also being able to simultaneously translate directly along its balance axis. Control is performed using a constrained time-optimal infinite horizon model predictive controller, with feasibility maintained across the full reference input set. Explored in this article is the derivation of the system dynamics model and controller, a systematic approach to selection of controller parameters and analysis of their effect on control performance and complexity, and an evaluation of the controller’s efficacy in both simulation and on a real-world experimental prototype for simple and complex trajectories.

Index Terms—Dynamics, Underactuated Robots, Wheeled Robots.

I. INTRODUCTION

THE Collinear Mecanum Drive (CMD) extends the mobility of a two-wheeled inverted pendulum (TWIP) to allow for omnidirectional movement, whilst simultaneously dynamically balancing in a single axis about the upright unstable equilibrium. This allows for the direct navigation of gaps smaller than the platform’s width, where a TWIP would have to perform a multi-point parallel parking manoeuvre due to the nonholonomic constraints imposed by its wheels. This is achieved using four individually actuated and suspended collinear Mecanum wheels, arranged in pairs with opposite handedness. Motion orthogonal to the balance axis is achieved by rotating all four wheels together as in a TWIP, whilst motion along the balance axis is generated by rotating wheels of opposite handedness against one another. This opposition of wheel rotation means that twice the total torque is required to achieve the same acceleration as a TWIP. Torque requirements are increased further when performing compound manoeuvres in both directions with a varying heading, meaning larger tractive forces between the wheels and ground are required than in a similar TWIP. This necessitates the constraining of

wheel torques in order to avoid wheel slip and loss of control. Furthermore, given a set of acceleration commands in these three dimensions that result in the motor torques reaching their constraints, a decision must be made as to how best to divide the available constrained wheel torque between the three competing manoeuvres, whilst still maintaining balance. Body velocity constraints must also be observed in order to ensure the generation of safe trajectories between distant references.

A prior work detailed the derivation of the kinematics and dynamics models for this wheel configuration [1]. Only this and a single other existing work explore the control of the CMD, in which the author implements cascaded manually tuned PID controllers [2].

The two-wheeled inverted pendulum, however, is well studied, and possesses similar dynamics to this platform. A number of review articles compare performance between different classical control techniques applied to the TWIP such as PID, LQR, and pole placement techniques. These typically find minimal performance difference between approaches given comparable parameter tuning, and consistently poor performance in the presence of constraints [3], [4]. Feedback linearisation can be used to negate some of the nonlinearities present in the system, achieved by a suitable change of variables and control input, transforming the nonlinear model into an equivalent linear one suitable for control by classical techniques. These methods have been applied to the TWIP for varying degrees of linearisation and control up to providing global position control for point to point manoeuvres [5]. Nonlinear optimal control has been implemented on a TWIP [6], using a nonlinear method similar to LQR to achieve full position control. However, none of these methods provide a systematic way of ensuring constraint satisfaction, and therefore must be provided with a suitable externally generated reference trajectory that results in the closed loop system observing the desired constraints. It must therefore be accepted that constraints may be violated during disturbance unless a suitable updated recovery trajectory can be provided sufficiently quickly as to prevent violation.

Model predictive control (MPC) uses a model of the plant to predict the response of the system to a number of successive control moves, which can be optimally chosen to reduce a cost function that defines the controller’s desired performance over a receding or infinite prediction horizon. This online optimisation allows for the systematic handling of constraints, making this type of controller well suited to this application. However, these approaches are much more computationally demanding due to their need to solve a numerical optimisation for every

*This work was supported by Consequential Robotics Ltd (CqR) and the Engineering and Physical Sciences Research Council [grant number EP/M508135/1]

^{1,2}M.T. Watson and D.T. Gladwin are with the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, UK. m.t.watson@sheffield.ac.uk, d.gladwin@sheffield.ac.uk

³T.J. Prescott is with the Department of Computer Science, University of Sheffield, Sheffield, UK. t.j.prescott@sheffield.ac.uk

⁴S.O. Conran is Director of Consequential Robotics, 2 Munden St, Hammersmith, London W14 0RH. sconran@consequentialrobotics.com

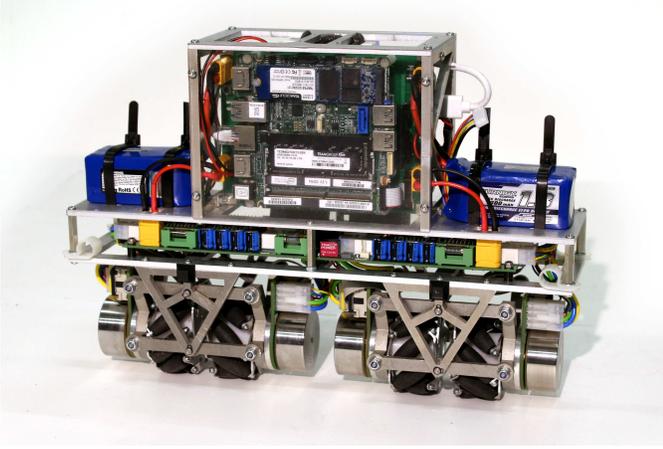


Fig. 1: Collinear Mecanum Drive prototype upon which modelling and experiments are based

control iteration, making their real-time implementation on systems with fast dynamics challenging. MPCs with linear prediction models and constraints have been well studied in the context of cart and pole inverted pendulum, but their application to TWIPs has been limited. Dini [7] implements a finite horizon MPC, but only for the control of the yaw and pitch states. Hirose [8] and Yue [9] also control the system's forward velocity, incorporating both pitch and velocity constraints. Both still rely on an externally generated velocity trajectory in order to avoid infeasibility when performing point-to-point manoeuvres. Ohhira [10] comes close to implementing a full MPC position controller, in which a stabilising inner LQR is used to provide a closed loop system that is then augmented by an outer optimal predictive controller, in a similar manner to dual-mode MPC. However, the inner loop is calculated with a 0.01 s sample time, whilst the outer optimiser is computed with a 0.08 s sample time, meaning that constraint satisfaction can only be guaranteed when the inner and outer loop sampling instants coincide. This intermittent enforcement of the hard input constraints could allow a large disturbance with an aggressive controller to demand a sufficiently large wheel torque as to induce wheel slip and loss of control.

This paper briefly recaps the derivation of the kinematics and dynamics model of the CMD, and then details the derivation, simulation, and implementation of a constrained dual-mode MPC on an experimental CMD prototype. This controller topology was chosen for its systematic design approach, its numerical simplicity in comparison to nonlinear MPC approaches, its *a priori* stability guarantee, and its improved numerical conditioning in the prediction of open loop unstable plants [11]. Additionally, recent approaches to ensuring feasibility for the full set of unconstrained reference inputs are incorporated whilst maintaining optimality [12], [13]. The effect of controller parameters on the feasible state set is analysed, and a systematic design approach is described for the selection of control horizon n_c and quadratic cost weighting matrices Q and R . Finally, both simulated and experimental results demonstrate the suitability and efficacy of this controller for the control of a CMD, performed using

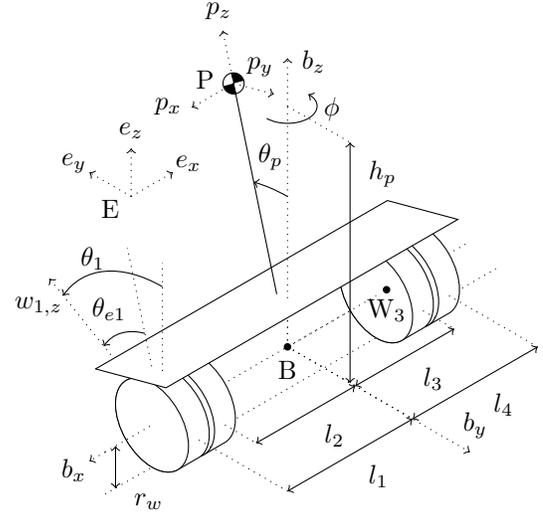


Fig. 2: Collinear Mecanum Drive coordinates and parameters

the prototype in Fig. 1.

II. MODEL DERIVATION

Consider the CMD depicted in Fig. 2 on a flat plane, where $\{E, e_x, e_y, e_z\}$ denotes the fixed reference frame, and $\{B, b_x, b_y, b_z\}$ the body attached frame obtained by rotating E about e_z by ϕ , with the origin of B located on the wheel rotation axis in the center of the platform, represented by the rotation matrix R_{eb} . $\{P, p_x, p_y, p_z\}$ represents the pendulum attached frame obtained by rotating B about b_x by θ_p , with rotation matrix R_{bp} . The origin of P is located at the pendulum center of mass, a translation of $h_p \hat{p}_z$ from B, with mass m_p and symmetric inertia tensor $I_p = \text{diag}(I_{px}, I_{py}, I_{pz})$. The wheel attached frames W_i for wheels $i = [1 \dots 4]$ are obtained by a translation of $l_i \hat{b}_x$ from B, and a rotation about b_x by θ_i , or by the rotation matrix R_{bw_i} . All wheels have identical masses m_w and symmetric inertia tensors $I_w = \text{diag}(I_{wx}, I_{wy}, I_{wz})$. Each wheel is of radius r_w , and has affixed about its circumference a ring of unactuated rollers, with their rotation axes offset from that of the parent wheel rotation axis and passing through the center of the roller. The contact point between the wheel and roller is assumed to be fixed directly under the center of the wheel. For this platform standard Mecanum wheels are used, so $\alpha_i = \pm\pi/4 \forall i$.

From [1] the nonholonomic no-slip constraint imposed by a Mecanum wheel can be defined in E as

$$\frac{\cos(\alpha + \phi)}{\sin(\alpha)} \dot{x} + \frac{\sin(\alpha + \phi)}{\sin(\alpha)} \dot{y} + \dot{\phi} l_i + r_w \dot{\theta}_i = 0 \quad (1)$$

This can be applied to wheels 1 through 4 and written in matrix form to define the platform's inverse kinematic mapping

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} = -\frac{1}{r_w} \begin{bmatrix} \frac{\cos(\alpha_1 + \phi)}{\sin(\alpha_1)} & \frac{\sin(\alpha_1 + \phi)}{\sin(\alpha_1)} & l_1 & 0 \\ \frac{\cos(\alpha_2 + \phi)}{\sin(\alpha_2)} & \frac{\sin(\alpha_2 + \phi)}{\sin(\alpha_2)} & l_2 & 0 \\ \frac{\cos(\alpha_3 + \phi)}{\sin(\alpha_3)} & \frac{\sin(\alpha_3 + \phi)}{\sin(\alpha_3)} & l_3 & 0 \\ \frac{\cos(\alpha_4 + \phi)}{\sin(\alpha_4)} & \frac{\sin(\alpha_4 + \phi)}{\sin(\alpha_4)} & l_4 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\theta}_p \end{bmatrix} \quad (2)$$

The system's dynamics equations are derived by use of the Euler-Lagrange equation in terms of the Lagrangian L , a column vector of generalised coordinates q and forces Q , a column vector of r Lagrange multipliers, and Pfaffian constraint matrix $M(q)$, defined as

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = Q + M^T(q)\lambda \quad (3)$$

where $M(q)$ satisfies $M(q)\dot{q} = 0$.

The generalised coordinates are selected as $q = [x, y, \phi, \theta_p, \theta_1, \theta_2, \theta_3, \theta_4]^T$, and $M(q)$ is derived from (2) as

$$M(q) = \begin{bmatrix} \frac{\cos(\alpha_1 + \phi)}{\sin(\alpha_1)} & \frac{\sin(\alpha_1 + \phi)}{\sin(\alpha_1)} & l_1 & 0 \\ \frac{\cos(\alpha_2 + \phi)}{\sin(\alpha_2)} & \frac{\sin(\alpha_2 + \phi)}{\sin(\alpha_2)} & l_2 & 0 \\ \frac{\cos(\alpha_3 + \phi)}{\sin(\alpha_3)} & \frac{\sin(\alpha_3 + \phi)}{\sin(\alpha_3)} & l_3 & 0 \\ \frac{\cos(\alpha_4 + \phi)}{\sin(\alpha_4)} & \frac{\sin(\alpha_4 + \phi)}{\sin(\alpha_4)} & l_4 & 0 \end{bmatrix} [r_w I_{4 \times 4}] \quad (4)$$

The Lagrangian L is found as the difference of system kinetic and potential energy as $L = K - U$, where K represents the sum of translational and rotational kinetic energy, and U the total potential energy. The rotational kinetic energy of the system is defined as the sum of rotational energy of the pendulum mass and four wheel masses as

$$K_r = \frac{1}{2} \vec{\omega}_p^T I_p \vec{\omega}_p + \frac{1}{2} \sum_{i=1}^4 \vec{\omega}_{w_i}^T I_w \vec{\omega}_{w_i} \quad (5)$$

where

$$\begin{aligned} \vec{\omega}_b &= \dot{\phi} \hat{b}_z \\ \vec{\omega}_p &= R_{bp}^T \vec{\omega}_b + \dot{\theta}_p \hat{p}_x \\ \vec{\omega}_{w_i} &= R_{b w_i}^T \vec{\omega}_b + \dot{\theta}_i \hat{w}_x \end{aligned} \quad (6)$$

Similarly, translational kinetic energy is defined as the sum of that of the pendulum and four wheel masses as

$$K_t = \frac{1}{2} \vec{v}_p^T m_p \vec{v}_p + \frac{1}{2} \sum_{i=1}^4 \vec{v}_{w_i}^T m_w \vec{v}_{w_i} \quad (7)$$

where

$$\begin{aligned} \vec{v}_p &= R_{bp}^T R_{eb}^T [\dot{x} \ \dot{y} \ 0]^T + \vec{\omega}_p \times h \hat{p}_z \\ \vec{v}_{w_i} &= R_{b w_i}^T R_{eb}^T [\dot{x} \ \dot{y} \ 0]^T + \vec{\omega}_{w_i} \times l_i \hat{w}_x \end{aligned} \quad (8)$$

Potential energy is purely that due to gravity as

$$U = m_p g h_p \cos(\theta_p) \quad (9)$$

and assuming no friction Q can be defined as

$$Q = \begin{bmatrix} 0_{1 \times 3} & \left(-\sum_{i=1}^4 \tau_i \right) & \tau_1 & \tau_2 & \tau_3 & \tau_4 \end{bmatrix}^T \quad (10)$$

where τ is a column vector in which τ_i represents a motor drive torque on wheel i . Note as the wheel actuators are mounted to the pendulum body a counter-torque is also applied to the pendulum.

Introducing Lagrange multipliers $\lambda \in \mathbb{R}^4$ allows the evaluation of (3), giving a system of eight ODEs. Defining Λ as a basis for the null space of M such that Λ annihilates M as $M\Lambda = 0$, a pre-multiplication of (3) by Λ^T allows the elimination of the Lagrange multipliers, reducing (3) to

Parameter	Value	Parameter	Value
α_1, α_4	$-\pi/4$ rad	I_{wx}	5.51×10^{-5} kg m ²
α_2, α_3	$\pi/4$ rad	I_{wyz}	5×10^{-5} kg m ²
m_p	2.47 kg	$l_1, -l_4$	0.105 m
m_w	0.145 kg	$l_2, -l_3$	0.063 m
I_{px}	0.0173 kg m ²	r_w	0.03 m
I_{py}, I_{pz}	0.025 kg m ²	h_p	0.055 m

TABLE I: Model parameters derived from a CAD model of the prototype in Fig. 1.

a system of four ODEs representing the dynamics of the system in terms of the reduced generalised coordinate vector $\zeta = [x \ y \ \phi \ \theta_p]^T$. These can be arranged into the standard passive Lagrangian form

$$M(\zeta)\ddot{\zeta} + C(\zeta, \dot{\zeta})\dot{\zeta} + G(\zeta) = H(\zeta)\tau \quad (11)$$

with symmetric positive definite inertia matrix $M(\zeta)$, centripetal and Coriolis matrix $C(\zeta, \dot{\zeta})$, chosen using the Christoffel symbols of $M(\zeta)$ so that $M - 2C$ is skew symmetric, gravity matrix $G(\zeta)$, and nonlinear input map $H(\zeta)$. Model parameters for the prototype shown in Fig. 1 are given in Table I.

III. CONTROL

Linearising (11) about the stationary upright position with $\phi = 0$ yields a prediction model suitable for development of a linear model predictive controller in the form

$$x_{k+1} = Ax_k + Bu_k \quad y_k = Cx_k \quad (12)$$

with state vector $x = [x \ y \ \phi \ \theta_p \ v_x \ v_y \ \dot{\phi} \ \dot{\theta}_p]$, in which v_x and v_y represent body frame velocities with time integrals x and y . This linearisation negates the nonlinearity in the integration of body accelerations to global positions with a varying ϕ ; correction of this error is assumed to be provided in the external generation of reference trajectories.

A dual-mode controller is chosen for its a priori stability guarantee and improved numerical conditioning in the prediction of open loop unstable plants [11]. This controller uses the control law

$$\begin{aligned} u_k &= -Kx_k + c_k & k \leq n_c \\ u_k &= -Kx_k & k > n_c \end{aligned} \quad (13)$$

where K represents an unconstrained optimal feedback, and c_k represents the first element of an optimised sequence of n_c future perturbations from the unconstrained optimal, denoted $\underline{c}_{\rightarrow k}$, where the notation $\underline{c}_{\rightarrow k}$ represents the column vector formed by stacking future values of c_j for $j = [k \dots k + n_c - 1]$, i.e. $\underline{c}_{\rightarrow k} = [c_k \ c_{k+1} \ \dots \ c_{k+n_c-1}]^T$.

A. Reference Tracking

In order to track a varying reference r_k the performance index applied at $y_k = r_k$ must propose no change to the input. To achieve this the model must be redefined in terms of deviations \hat{x}_k and \hat{u}_k from the desired steady states $x_{ss|k}$ and $u_{ss|k}$ at step k as $x_k = \hat{x}_k + x_{ss|k}$ and $u_k = \hat{u}_k + u_{ss|k}$.

The steady state values are calculated by the simultaneous equations $y_{ss|k} = Cx_{ss|k}$, $x_{ss|k} = Ax_{ss|k} + Bu_{ss|k}$, arranged to solve for $x_{ss|k}$, $u_{ss|k}$ in matrix form with $y_{ss|k} = r_k$ as

$$\begin{bmatrix} C & 0 \\ A - I & B \end{bmatrix}^+ \begin{bmatrix} r \\ 0 \end{bmatrix} = \begin{bmatrix} x_{ss|k} \\ u_{ss|k} \end{bmatrix} = \begin{bmatrix} M_x \\ M_u \end{bmatrix} \begin{bmatrix} r \\ 0 \end{bmatrix} \quad (14)$$

allowing $x_{ss|k}$, $u_{ss|k}$ to be defined as

$$x_{ss|k} = M_x r_k \quad u_{ss|k} = M_u r_k \quad (15)$$

Substituting (13) into (12) and applying the above change of variables gives

$$\begin{aligned} x_{k+1} - x_{ss|k+1} &= \Phi(x_k - x_{ss|k}) + Bc_k \\ u_k - u_{ss|k} &= -K(x_k - x_{ss|k}) + c_k \end{aligned} \quad (16)$$

where $\Phi = A - BK$, which can then be substituted with (15) to give

$$\begin{aligned} x_{k+1} &= \Phi x_k + (I - \Phi)M_x r_{k+1} + Bc_k \\ u_k &= -Kx_k + (KM_x + M_u)r_{k+1} + c_k \end{aligned} \quad (17)$$

B. Autonomous Model Formulation & Reference Previewing

Instead of continuing with two separate modes as in (13), analysis can be simplified by the formation of an all-encompassing autonomous prediction model of the form $Z_{k+1} = \Psi Z_k$. This also allows constraints of the form $Gx \leq f$ to be applied to predicted step Z_{k+n} as $G\Psi^n Z_k \leq f$.

Additionally, advanced knowledge of n_r future reference inputs can be incorporated by the inclusion of $\underline{r}_{\rightarrow k+1} = [r_{k+1} \ r_{k+2} \ \dots \ r_{k+n_r+1}]^T$ in the autonomous model state. Typically $n_r \leq n_c$, as designing a controller to optimise its trajectory against a longer reference previewing horizon than its control horizon introduces a transient tracking error and yields undesirable performance [11]. In this article it is assumed that $n_r = n_c$. The change in advanced knowledge available to the controller over k can be defined as

$$\underbrace{\begin{bmatrix} r_{k+2} \\ r_{k+3} \\ \vdots \\ r_{k+n_r+1} \\ r_{k+n_r+1} \end{bmatrix}}_{\underline{r}_{\rightarrow k+2}} = \underbrace{\begin{bmatrix} 0 & I & 0 & \dots & 0 \\ 0 & 0 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & I \end{bmatrix}}_{D_r} \underbrace{\begin{bmatrix} r_{k+1} \\ r_{k+2} \\ \vdots \\ r_{k+n_r+1} \end{bmatrix}}_{\underline{r}_{\rightarrow k+1}} \quad (18)$$

in which at $k > n_r$ the controller's final steady state reference is assumed to be equal to r_{k+n_r+1}

Similarly, the matrix D_c captures the change in the future perturbation vector $\underline{c}_{\rightarrow k}$ over one prediction step

$$\underbrace{\begin{bmatrix} c_{k+1} \\ \vdots \\ c_{k+n_c} \\ 0 \end{bmatrix}}_{\underline{c}_{\rightarrow k+1}} = \underbrace{\begin{bmatrix} 0 & I & 0 & \dots & 0 \\ 0 & 0 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{D_c} \underbrace{\begin{bmatrix} c_k \\ c_{k+1} \\ \vdots \\ c_{k+n_c} \end{bmatrix}}_{\underline{c}_{\rightarrow k}} \quad (19)$$

This allows the definition of the autonomous prediction model Ψ as

$$\underbrace{\begin{bmatrix} x_{k+1} \\ \underline{c}_{\rightarrow k+1} \\ r_{\rightarrow k+2} \end{bmatrix}}_{Z_{k+1}} = \underbrace{\begin{bmatrix} \Phi & BD_{c_k} & (\Phi - I)M_x D_{r_{k+1}} \\ 0 & D_c & 0 \\ 0 & 0 & D_r \end{bmatrix}}_{\Psi} \underbrace{\begin{bmatrix} x_k \\ \underline{c}_{\rightarrow k} \\ r_{\rightarrow k+1} \end{bmatrix}}_{Z_k} \quad (20)$$

where D_{c_k} and $D_{r_{k+1}}$ select the first elements of $\underline{c}_{\rightarrow k}$ and $\underline{r}_{\rightarrow k+1}$ respectively.

C. Cost Function Derivation

A standard quadratic cost function that drives the state and input towards their desired steady state values can be defined as

$$\begin{aligned} J &= \sum_{i=1}^{\infty} (x_{k+i} - x_{ss|k+i})^T Q (x_{k+i} - x_{ss|k+i}) \\ &\quad + (u_{k+i-1} - u_{ss|k+i-1})^T R (u_{k+i-1} - u_{ss|k+i-1}) \end{aligned} \quad (21)$$

where Q and R are diagonal matrices scaling the quadratic cost of each state and input error.

$x_{k+i} - x_{ss|k+i}$ and $u_{k+i-1} - u_{ss|k+i-1}$ can be redefined in terms of the autonomous model state Z_k as

$$x_{k+i} - x_{ss|k+i} = \underbrace{\begin{bmatrix} I & 0 & -M_x D_{r_{k+1}} \end{bmatrix}}_{K_{xss}} Z_{k+i-1} \quad (22)$$

$$u_{k+i-1} - u_{ss|k+i-1} = \underbrace{\begin{bmatrix} -K \\ D_{c_k} \\ -M_u D_{r_{k+1}} \end{bmatrix}^T}_{K_{uss}} Z_{k+i-1} \quad (23)$$

allowing (21) to be rewritten in terms of K_{xss} , K_{uss} , and Z_k as

$$\begin{aligned} J &= \sum_{i=1}^{\infty} (K_{xss} Z_{k+i})^T Q (K_{xss} Z_{k+i}) \\ &\quad + (K_{uss} Z_{k+i-1})^T R (K_{uss} Z_{k+i-1}) \end{aligned} \quad (24)$$

which can be rewritten as

$$\begin{aligned} J &= \sum_{i=0}^{\infty} (K_{xss} Z_{k+i+1})^T Q (K_{xss} Z_{k+i+1}) \\ &\quad + (K_{uss} Z_{k+i})^T R (K_{uss} Z_{k+i}) \end{aligned} \quad (25)$$

It is then possible to substitute for Z_{k+i+1} using $Z_{k+i+1} = \Psi Z_{k+i}$

$$\begin{aligned} J &= \sum_{i=0}^{\infty} Z_{k+i}^T [(K_{xss} \Psi)^T Q (K_{xss} \Psi) \\ &\quad + (K_{uss})^T R (K_{uss})] Z_{k+i} \end{aligned} \quad (26)$$

From the autonomous model it is then seen that $Z_{k+i} = \Psi^i Z_k$, allowing substitution of Z_{k+i} and factorisation of the now constant Z_k term

$$\begin{aligned} J &= Z_k^T \left\{ \sum_{i=0}^{\infty} (\Psi^i)^T \left(K_{xss}^T \Psi^T Q K_{xss} \Psi \right. \right. \\ &\quad \left. \left. + K_{uss}^T R K_{uss} \right) \Psi^i \right\} Z_k \end{aligned} \quad (27)$$

The discrete Lyapunov equation can then be used to convert this convergent infinite series into the form

$$J = Z_k^T S Z_k = \begin{bmatrix} x_k \\ \underline{c}_k \\ r_{k+1} \end{bmatrix}^T \begin{bmatrix} S_x & S_{xc} & S_{xr} \\ S_{xc}^T & S_c & S_{cr} \\ S_{xr}^T & S_{cr}^T & S_r \end{bmatrix} \begin{bmatrix} x_k \\ \underline{c}_k \\ r_{k+1} \end{bmatrix} \quad (28)$$

This can be minimised by the solution of

$$\frac{dJ}{d\underline{c}_k} = 0 \quad (29)$$

to give the unconstrained optimal perturbation required to incorporate advanced reference knowledge

$$\underline{c}_k = -S_c^{-1} (S_{xc}x_k + S_{cr}r_{k+1}) \quad (30)$$

As expected, it is found that $\underline{c}_k = [0 \ 0 \ \dots \ 0]$ for $r_{k+1} = [r_{k+1} \ r_{k+1} \ \dots \ r_{k+1}]^T$, and $S_{xc} \equiv 0$ and can therefore be omitted [11].

It is therefore clear that $\underline{c}_k \neq 0$ if any advanced reference knowledge is included, even for the unconstrained case. This goes against the dual-mode paradigm, in that for the unconstrained case the optimal trajectory should be fully captured by the underlying control law, meaning that in the absence of constraints we should have $\underline{c}_k = [0 \ 0 \ \dots \ 0]^T \forall r_{k+1}$. To reinstate this intrinsic unconstrained optimality, the perturbation term can be redefined as $\underline{c}_k = \hat{\underline{c}}_k + -S_c^{-1}S_{cr}r_{k+1}$, now instead optimising for $\hat{\underline{c}}_k$. The elements not dependant on the DoF $\hat{\underline{c}}_k$ can then be removed, allowing the redefinition of the QP and control law (17) as

$$\min_{\hat{\underline{c}}_k} J = \begin{bmatrix} \hat{\underline{c}}_k - S_c^{-1}S_{cr}r_{k+1} \end{bmatrix}^T S_c \begin{bmatrix} \hat{\underline{c}}_k - S_c^{-1}S_{cr}r_{k+1} \end{bmatrix} + 2 \begin{bmatrix} \hat{\underline{c}}_k - S_c^{-1}S_{cr}r_{k+1} \end{bmatrix}^T S_{cr}r_{k+1} \quad (31)$$

$$u_k = -Kx_k + [(KM_x + M_u)D_{r_{k+1}} - D_{c_k}S_c^{-1}S_{cr}]r_{k+1} + \hat{c}_k \quad (32)$$

This new control law is presented as a control block diagram in Fig. 3. All blocks execute at the same discrete interval. Since the unconstrained optimal perturbation is known to be $\hat{\underline{c}}_k = 0$, the performance index must be purely quadratic [14], allowing (31) to be simplified to

$$\min_{\hat{\underline{c}}_k} J = \hat{\underline{c}}_k^T S_c \hat{\underline{c}}_k \quad (33)$$

The autonomous model (20) can be redefined to incorporate this reference previewing feedforward as

$$\begin{bmatrix} x_{k+1} \\ \underline{c}_{k+1} \\ r_{k+2} \end{bmatrix} = \underbrace{\begin{bmatrix} \Phi & BD_{c_k} & \Gamma \\ 0 & D_c & 0 \\ 0 & 0 & D_r \end{bmatrix}}_{\Psi} \underbrace{\begin{bmatrix} x_k \\ \underline{c}_k \\ r_{k+1} \end{bmatrix}}_{Z_k} \quad (34)$$

where $\Gamma = -BD_{c_k}S_c^{-1}S_{cr} + (I - \Phi)M_xD_{r_{k+1}}$.

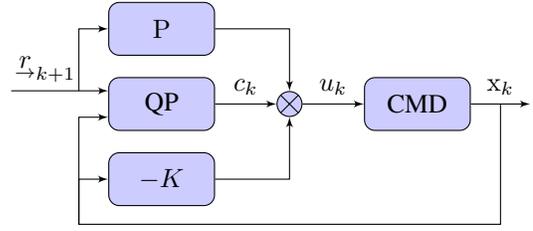


Fig. 3: Control block diagram, with feedforward matrix $P = (KM_x + M_u)D_{r_{k+1}} - D_{c_k}S_c^{-1}S_{cr}$.

D. Infeasible Reference Tracking

For this controller to be practically useful it must be able to drive the platform to any reference position and heading. However, the nature of the dual mode controller means that there must exist a feasible control trajectory \underline{c}_k that takes the platform into the set of states from which the closed loop system at $k > n_c$ will not violate any constraints over the infinite horizon, referred to as the maximal admissible set (MAS, \mathcal{S}_{MAS}) [11]. This lies within a superset of states from which there exists a sequence of feasible control moves \underline{c}_k that drive the initial state into the MAS, referred to as the maximal controlled admissible set (MCAS, \mathcal{S}_{MCAS}). For any initial state outside \mathcal{S}_{MCAS} there does not exist a sequence of control moves that can move the state into \mathcal{S}_{MAS} within $k \leq n_c$ without violating constraints, rendering the QP infeasible. For this application this limits step translations to approximately 0.1m for $n_c = 10$, dependant on initial state and choice of quadratic cost function matrices Q and R , rendering the controller impractical for real-world implementation.

Multiple approaches exist to addressing this problem. Simon [15] replaces r with a pseudo-reference \tilde{r} as an additional degree of freedom, penalising deviation of this from the true reference. Dughman [13] introduces an extra perturbation term c_∞ to the end of the control sequence \underline{c}_k that acts as a constant perturbation to the input for $k > n_c$. This constant perturbation has the same effect as a pseudo-reference, with the equivalent pseudo-reference calculable as $\tilde{r} = C(I - \Phi)^{-1}Bc_\infty + r$.

Here the latter approach is taken, introducing a c_∞ term for $k > n_c$. The opportunity is also taken to introduce a vector of slack variables \underline{s}_k that will be later used to soften the controller's output constraints, giving an updated autonomous model Ψ and state Z

$$\underbrace{\begin{bmatrix} x_{k+1} \\ \underline{c}_{k+1} \\ c_\infty \\ r_{k+2} \\ \underline{s}_{k+1} \end{bmatrix}}_{Z_{k+1}} = \underbrace{\begin{bmatrix} \Phi & BD_{c_k} & 0 & \Gamma & 0 \\ 0 & D_c & E_c & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & D_r & 0 \\ 0 & 0 & 0 & 0 & D_s \end{bmatrix}}_{\Psi} \underbrace{\begin{bmatrix} x_k \\ \underline{c}_k \\ c_\infty \\ r_{k+1} \\ \underline{s}_k \end{bmatrix}}_{Z_k} \quad (35)$$

where E_c is used to replace the last value of \underline{c}_{k+1} with c_∞ as $E_c = [0 \ 0 \ \dots \ 0 \ I]^T$, and where D_s represents a shift matrix with a similar structure as D_c . \underline{s}_k is chosen to be much larger than required for the derivation of constraints, with unused slack variables later removed by the trimming of empty columns from F_s and their associated element in S_s .

The QP (33) can now be redefined to additionally optimise for c_∞ and \underline{s}_k as

$$\min_{\{\hat{c}_{\rightarrow k}, c_\infty, \underline{s}_k\}} J = \begin{bmatrix} \hat{c}_{\rightarrow k} \\ c_\infty \\ \underline{s}_k \end{bmatrix}^T \begin{bmatrix} S_c & 0 & 0 \\ 0 & W S_{c_\infty} & 0 \\ 0 & 0 & S_s \end{bmatrix} \begin{bmatrix} \hat{c}_{\rightarrow k} \\ c_\infty \\ \underline{s}_k \end{bmatrix} \quad (36)$$

S_{c_∞} is scaled by a very small factor W so that choice of c_∞ has a minimal effect on $\hat{c}_{\rightarrow k}$, whilst ensuring $c_\infty \rightarrow 0$ as $k \rightarrow \infty$ and feasibility $\forall r_{\rightarrow k+1}$. S_s is a diagonal matrix of large slack weights used to heavily penalise deviation of \underline{s}_k from 0, so that its elements are only optimised to be substantially larger than zero if feasibility of the QP would be otherwise lost.

E. Selection of Q , R , T_s

The discretisation period T_s must be chosen to trade off disturbance rejection and tracking performance against robustness to solver delay and computational simplicity, and is chosen through trial and error as $T_s = 35$ ms.

For this application the controller is desired to track reference body positions and heading, so Q is initially set to $Q = \text{diag}([1 \ 1 \ 1 \ 0_{1 \times 5}])$. R must be chosen as a trade-off between control performance and both robustness to estimation error and prediction uncertainty, so a value of $R = 0.1I_{4 \times 4}$ is chosen.

F. Constraint Derivation

Hard input constraints on the wheel torques $|\tau| \leq \bar{\tau}$ are required in order to avoid wheel slip. Output constraints are required on the θ_p state in order to prevent the controller from attempting to translate using an unrealistic lean angle, as well as to keep the system near the model operating point. The v_x , v_y and $\dot{\phi}$ states must be constrained in order to maintain a safe margin from the edge of \mathcal{S}_{MCAS} for the controller to be able to handle disturbances, as well as to ensure the controller generates safe and sensible velocity profiles.

Hard constraints on the input $|u_k| \leq \bar{u}$ and softened output constraints $|x_k| \leq \bar{x}$ at time k are represented in the form

$$\underbrace{\begin{bmatrix} -K & D_{c_k} & 0 & P & 0 \\ K & -D_{c_k} & 0 & -P & 0 \\ C & 0 & 0 & 0 & -D_{s_k} \\ -C & 0 & 0 & 0 & -D_{s_k} \end{bmatrix}}_G \underbrace{\begin{bmatrix} x_k \\ \hat{c}_{\rightarrow k} \\ c_\infty \\ r_{\rightarrow k+1} \\ \underline{s}_k \end{bmatrix}}_{Z_k} \leq \underbrace{\begin{bmatrix} \bar{u} \\ \bar{u} \\ \bar{x} \\ \bar{x} \end{bmatrix}}_f \quad (37)$$

where $P = -D_{c_k} S_c^{-1} S_{c_r} + (K M_x + M_u) D_{r_k}$.

These constraints can be projected over an n_{con} constraint horizon by use of the autonomous model as

$$\underbrace{\begin{bmatrix} G \\ G\Psi \\ G\Psi^2 \\ \vdots \\ G\Psi^{n_{con}} \end{bmatrix}}_F Z_k \leq \underbrace{\begin{bmatrix} f \\ f \\ \vdots \\ f \end{bmatrix}}_t \quad (38)$$

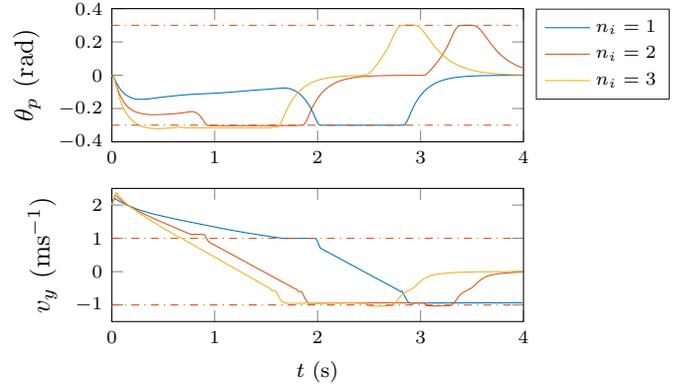


Fig. 4: Comparison of controller response in the regulation case to an initial constraint violation of $v_y = 2\bar{v}_y$ for $n_i = [1, 2, 3]$

where $n_{con} > n_c$, and n_{con} is sufficiently large as to fully define \mathcal{S}_{MAS} , as from Section III-D membership of \mathcal{S}_{MAS} at $k = n_c + 1$ is sufficient to guarantee constraint satisfaction over the infinite horizon. However, this approach to defining \mathcal{S}_{MAS} can result in redundant constraints, and provides no systematic method for selection of n_{con} .

Fortunately, algorithms for deriving the minimal set of constraints required to capture \mathcal{S}_{MAS} are well studied in the literature [16] [17], so these existing methods are utilised to derive F and are therefore not discussed any further¹.

G. Slack Variable Distribution

For the prototype described in this paper with $n_c = 10$, \mathcal{S}_{MAS} can be defined using approximately 90 output constraints. Softening each of these constraints individually requires an equal number of slack variables, increasing the QP DoF from 52 to 142, a large increase in complexity. To lessen this, the number of slack variables can be reduced by sharing slack variables across multiple timesteps through redefinition of F_s and removal of entries from S_s . For this controller individual slack variables are used for the first n_i timesteps, with $n_i = 1$ for the v_x and $\dot{\phi}$ states and $n_i = 3$ for the θ_p and v_y states, with a single slack variable per state shared for the rest of the infinite horizon. This reduces n_s from $n_s \approx 90$ to $n_s = 12$. While this method with $n_i = 1$ for all states is sufficient to maintain feasibility during disturbance, the controller has insufficient degrees of freedom to quickly address constraint violations. A comparison of controller response for $n_i = [1, 2, 3]$ in the regulation case to an initial disturbance of $v_y = 2\bar{v}_y$ is shown in Fig. 4, in which a large variation in the time taken to re-establish constraint satisfaction is apparent between different values of n_i .

Slack weights in (36) for non-shared slack variables are set to near zero to allow the controller to worsen constraint violation over a short horizon to improve the rate of convergence to S_O over the infinite horizon. For example, for a system with $v_y = 2\bar{v}_y$ the desirable response is for the

¹Elements of F and Ψ that multiply onto $\underline{s}_{\rightarrow k}$ must be temporarily removed whilst deriving the MAS.

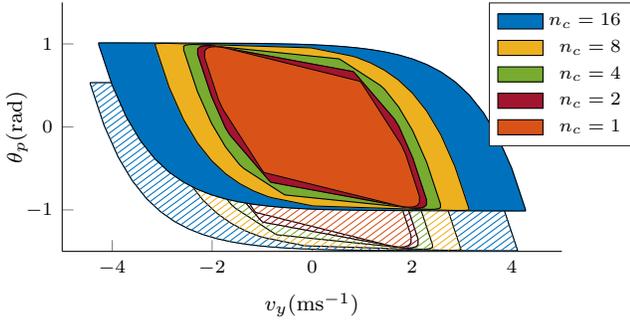


Fig. 5: Cross section of \mathcal{S}_{MCAS} across θ_p and v_y , taken through the origin (solid, front) and through $\dot{\theta}_p = 4 \text{ rad s}^{-1}$ (hatched, back), with varying control horizon n_c and $Q = \text{diag}(1_{1 \times 3}, 0_{1 \times 5})$.

controller to briefly increase v_y in order to drive θ_p negative to start decelerating. A large slack weight for the slacks at $k \leq 3$ penalises this type of quick correction, preventing the system from correcting the violation as aggressively as is desired, resulting in a longer duration of constraint violation. The infinite horizon slack weights are set to the region of 10^5 to ensure that all resulting trajectories bring the state towards S_O as quickly as possible, regardless of reference. The cost of s_∞ must also be much greater than that due to tracking error, otherwise a distant reference trajectory will cause the controller to purposefully generate a constraint violation.

H. Constraint Feasibility

While the output constraints applied to v_x , v_y , $\dot{\phi}$ and θ_p are softened to maintain feasibility during disturbance, the hard input constraints applied to u_k in combination with the terminal constraint set effectively apply their own constraints to v_x , v_y and θ_p , albeit with a larger constrained range. This is due to the requirement for the controller to direct the state into \mathcal{S}_{MAS} at $k = n_c + 1$. For example, for a system with a large v_y and θ_p of the same sign, a large control horizon is required in order to give the controller enough time to manipulate the plant into \mathcal{S}_{MAS} using its constrained input. This can lead to parts of the output constraint set S_O being infeasible. Care must therefore be taken to ensure n_c and \bar{u} are sufficiently large when specifying S_O in order to ensure $S_O \subseteq \mathcal{S}_{MCAS}$. Additionally, a large margin must be allowed due to use of softened output constraints, otherwise an acceptable constraint violation could result in infeasibility. For this application the maximum possible v_y and θ_p values for which the controller must remain feasible are expected to be $\pm 3 \text{ m s}^{-1}$ and $\pm 0.4 \text{ rad}$ respectively, with minimal disturbance expected in the open-loop stable v_x and $\dot{\phi}$ states.

The implied constraint for each element of x can be found by fixing all other elements of x and solving a linear program to find the maximum constrained value of the remaining element of x . For example, the maximum static lean angle can be found by

$$\max_{\{\theta_p, \underline{c}_k, c_\infty\}} [1 \ 0 \ \dots \ 0] \begin{bmatrix} \theta_p \\ \underline{c}_k \\ c_\infty \end{bmatrix} \quad s.t. \quad F \begin{bmatrix} \theta_p \\ \underline{c}_k \\ c_\infty \end{bmatrix} \leq t \quad (39)$$

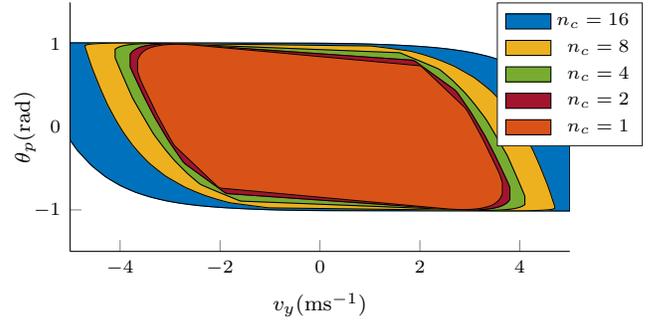


Fig. 6: Cross section of \mathcal{S}_{MCAS} across θ_p and v_y for varying control horizon n_c and $Q = \text{diag}(1_{1 \times 3}, 0_{1 \times 4}, 0.01)$.

where F and t are redefined to include the now fixed elements of x .

Analysing a 2D cross-section of \mathcal{S}_{MCAS} at the origin across the v_y and θ_p states shows the relationship between feasibility and these two states, shown in Fig. 5. Interestingly, for the QP to remain feasible for the desired output constraint set and anticipated magnitude of constraint violation a control horizon of $n_c \geq 8$ is required, placing a lower limit on n_c for this set of controller parameters. This analysis also ignores the effect of the other remaining states on feasibility, with the hatched areas in Fig. 5 showing a cross section of \mathcal{S}_{MCAS} through $\dot{\theta}_p = 4 \text{ rad s}^{-1}$, a more realistic representation of a large external impulse disturbance along the b_y axis. This emphasises the importance of a large control horizon and therefore large MCAS if a combination of equally signed disturbances in the v_y , θ_p , and $\dot{\theta}_p$ states are to not result in infeasibility. The relationship between choice of n_c and \bar{u} also extends to the x and ϕ subsystems, but neither require a long control horizon in order for the MCAS to encompass the desired output constraints and their anticipated violations.

These figures demonstrates one of the main disadvantages of the dual-mode approach to predictive control, in that for a given cost function a long control horizon can be required in order to ensure a sufficiently large MCAS relative to the soft output constraint set. For the same cost function and constraints this can only be addressed by increasing n_c at a cost of execution time, or by increasing T_s at a cost of control performance and model accuracy.

Alternatively, a modification to the cost function can be used to manipulate the unconstrained feedback K such that a larger MCAS can be obtained for the same n_c and T_s . Q is therefore modified to $Q = \text{diag}(1_{1 \times 3}, 0_{1 \times 4}, 0.01)$, with a cross-section of the new MCAS shown in Fig. 6. This shows that a small control horizon is now able to fully access the anticipated constraint violation set, at a cost of less aggressive control of the y state. A similar effect could be achieved through control move blocking [18], however, this method does not allow the embedding of the reference previewing feedforward term into the unconstrained control law as in (32).

IV. SIMULATION RESULTS

The quadratic program is solved using qpOASES [19], an online active set solver, implemented on an Intel i7-

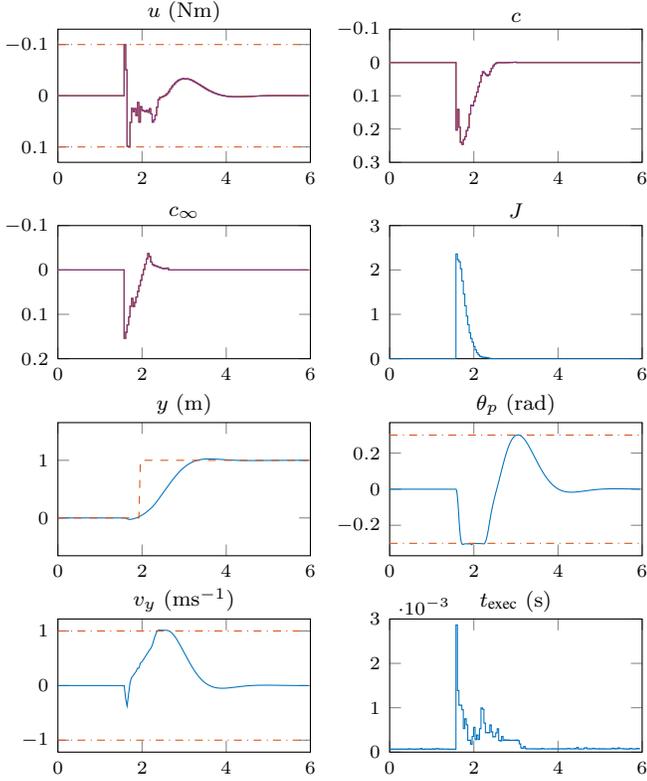


Fig. 7: Simulated controller response to a step reference of $y = [0, 1]m$. Trajectories are solid lines, references are dashed, constraints are dot-dash.

4720HQ processor for simulated results and an Intel i7-8650U for experimental results. Simulation is performed using the continuous time nonlinear plant model.

A. Step Reference Tracking

Fig. 7 shows the simulated response of the controller to a step reference input of $r_y = [0, 1]$. This shows a response with minimal overshoot and sensible preemption of the reference change. c_∞ correctly increases at the moment of the reference step to maintain feasibility, tending to zero as the system approaches the reference. The controller shows satisfaction of all constraints, saturating the input for a number of samples and producing near minimum-time \dot{v}_y and θ_p trajectories. Execution time peaks at the instant of the reference step to 3 ms, an acceptable control delay, quickly dropping below 1 ms for the remainder of the trajectory, and computing nearly instantly once $x \in \mathcal{S}_{MAS}$.

Fig. 8 shows the system's response to an $r_x = [0, 1]$ step reference input. This demonstrates close to bang-bang control without any constraint violation, again demonstrating the ability of the optimal control approach to produce close to minimum time trajectories in the presence of constraints. A step reference applied to r_ϕ generates very similar trajectories and is therefore omitted.

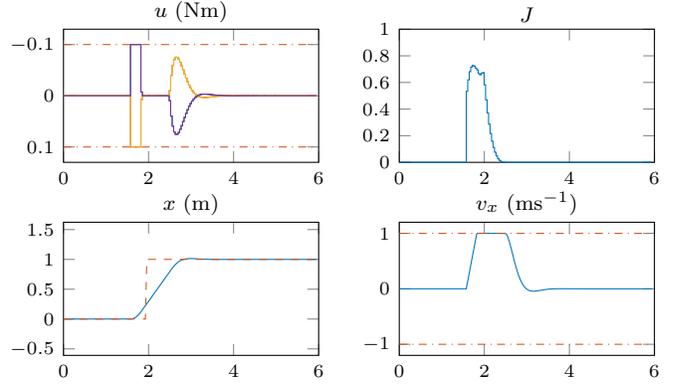


Fig. 8: Simulated controller response to a step reference of $r_x = [0, 1]m$

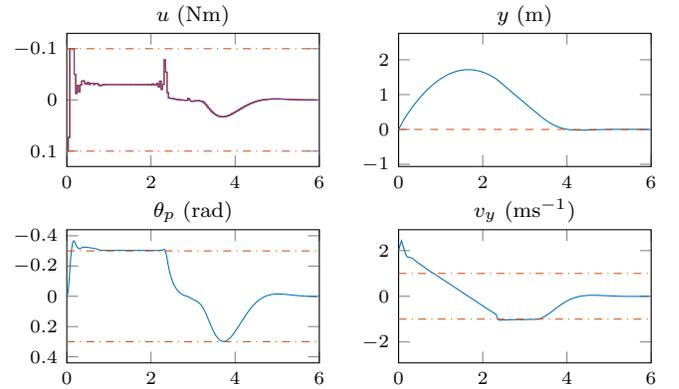


Fig. 9: Simulated controller response in regulation scenario to an initial disturbance of $v_y = 2 \text{ m s}^{-1} = 2\bar{v}_y$

B. Constraint-Violating Disturbance Handling

Fig. 9 aims to demonstrate the controller's handling of large constraint-violating disturbances. For this figure the reference is kept at the origin and the system is given an initial forward velocity $v_y = 2 \text{ m s}^{-1}$, whilst maintaining all other initial states the origin. The controller is seen to briefly worsen constraint violation by increasing v_y to lean the platform towards the origin, maintaining $\theta_p = \bar{\theta}_p$ to decelerate as quickly as constraints allow, before maintaining $v_y = -\bar{v}_y$ until $x = 0$ with no overshoot.

C. Figure-of-8 Trajectory Tracking

In Fig. 10 a figure-of-8 trajectory of 10 s duration is used to demonstrate the response of the controller to a more complex trajectory for $n_c = 10$, in which a small phase lag is evident in the y state. This highlights a drawback of the dual-mode MPC approach; the use of a closed loop prediction model means that when given a previewed section of a continuously changing reference the controller is optimising its future trajectory to come to rest at r_{k+n_c+1} , meaning it must plan for its state at $k = n_c + 1$ to lie within \mathcal{S}_{MAS} . In practise this means that the system can only increase v_y to a value from which it can enter \mathcal{S}_{MAS} in n_c timesteps, which for the value of n_c used here is insufficient to correctly track the given velocity profile. This

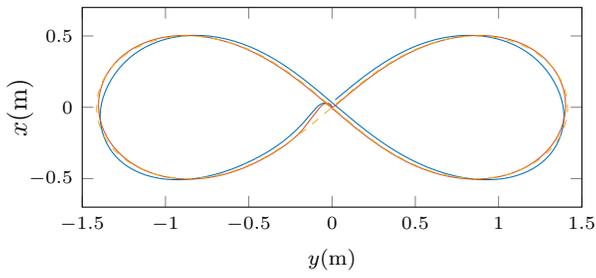


Fig. 10: Simulated state trajectories with $n_c = 10$ (blue) and $n_c = 28$ (red) for a figure-of-8 reference with constant ϕ (yellow, dashed), starting at the origin with an initial direction of down and left.

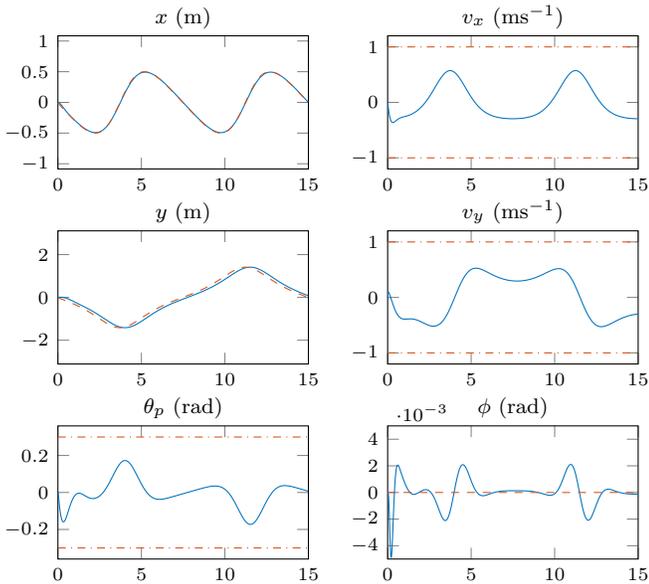


Fig. 11: State trajectories for Fig. 10 with $n_c = 10$.

results in the system lagging behind the desired trajectory, which in turn increases the effective stopping distance from the target steady state that the controller is able to use to enter \mathcal{S}_{MAS} at $k = n_c + 1$ from a given velocity. This phase lag increases until the system has accumulated sufficient distance from r_{k+n_c+1} to be able to safely reach the velocity required to keep up with the moving reference.

This can be addressed by ensuring a reference previewing period that is of sufficient length to fully capture the transition of the system from any state within \mathcal{S}_O into \mathcal{S}_{MAS} without constraint violation. This value of n_c can be approximated by examining the system response in the regulation case to an initial forward velocity of $v_y = 1$, which indicates a stopping distance of 0.53 m, taking 0.56 s for the system to enter \mathcal{S}_{MAS} . Two approaches exist to ensure this; the reference previewing period and therefore n_c can be increased to match the stopping time at $n_c = 28$, at a cost of greater computational complexity, or the underlying gain K can be increased and the output constraint set enlarged in order to allow the system to reach the required steady state in less time. However, increasing K in turn decreases the size of \mathcal{S}_{MCAS} as discussed in Section

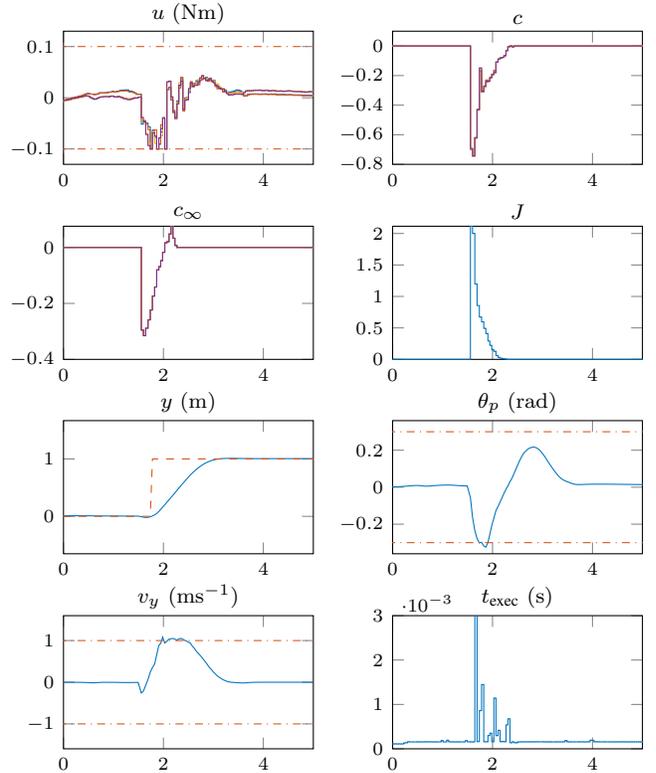


Fig. 12: Experimental response to a step reference of $r_y = [0, 1]$ m

III-H, as well as increasing sensitivity to parameter/model uncertainty and noise. Enlarging the output constraint set is also ineffective, as the same problem occurs, only at a larger velocity. For demonstration n_c and n_r are therefore increased to 28, with the response to the same figure-of-8 trajectory also shown in Fig. 10. This also allows a reduction of R in contrast to Section III-H, as this larger control horizon already enlarges \mathcal{S}_{MCAS} to encompass \mathcal{S}_O with sufficient margin. However, this increase in control horizon increases worst case execution time to $t_{exec} = 6.4$ ms, which in practise is too large for the real-time control of this particular system. The control horizon is therefore left unchanged, and a small amount of transient tracking error at large velocities is accepted.

Also visible in Fig. 11 is a cross coupling between the v_x and ϕ states. This is due to the use of a linearised prediction model, but is a minor error with a peak of 2×10^{-2} rad. This, along with good constraint tracking of θ_p in Fig. 7 and 9 indicates that the linearised model is a suitably accurate approximation of the real system for use as a prediction model with this choice of \mathcal{S}_O .

V. EXPERIMENTAL RESULTS

The same step and figure-of-8 trajectory references from Fig. 7-11 are now applied to the experimental prototype. The controller output is updated as continuously as execution time allows in order to improve disturbance rejection, with a typical control update rate of 10 ms.

Fig. 12 shows the prototype's response to a step of $r_y = [0, 1]$ m. The resulting trajectory has a good similarity to that

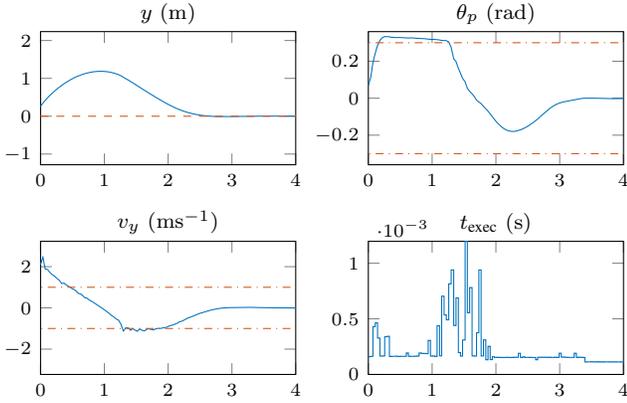


Fig. 13: Experimental response in the regulation case to an initial velocity disturbance of approximately 2 m s^{-1}

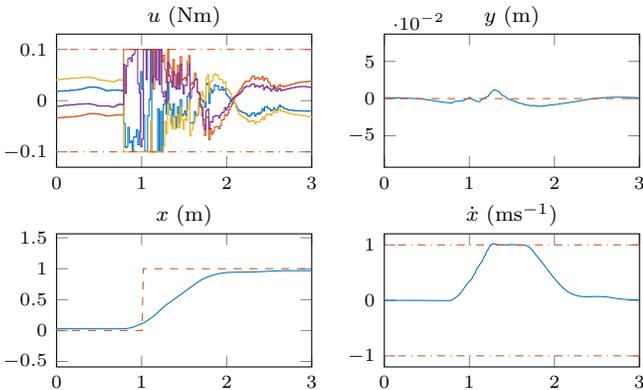


Fig. 14: Experimental response to a step reference of $r_x = [0, 1] \text{ m}$

in Fig. 7, showing acceptable satisfaction of input and output constraints. Fig. 13 shows the prototype's response to a large constraint violating velocity disturbance of $v_y \approx 2 \text{ m s}^{-1} = 2\bar{v}_y$ as in Fig. 9, again showing very similar performance to that predicted by simulation.

Fig. 14 shows a step input in r_x . This demonstrates the benefit of using a constrained optimal controller to directly control motor torques, in that the controller is able to optimally saturate the input for a large number of timesteps with no negative impact on controller stability. This figure also shows minimal disturbance to the y state when performing lateral movement, and shows exact constraint satisfaction. This response does exhibit a small steady-state error, likely due to the stable damped dynamics of this subsystem and static friction in the Mecanum wheel roller bearings. This could be addressed in future work by the incorporation of integral action into the controller, likely through an output disturbance observer [11], or by incorporation of friction into the prediction model.

Finally, Fig. 15 shows the prototype's response to the same figure-of-8 trajectory as in Fig. 11. This shows the same expected phase lag in the y state as in simulation, along with the same x error as in Fig. 14, now also presenting as small degree of lag behind the desired trajectory.

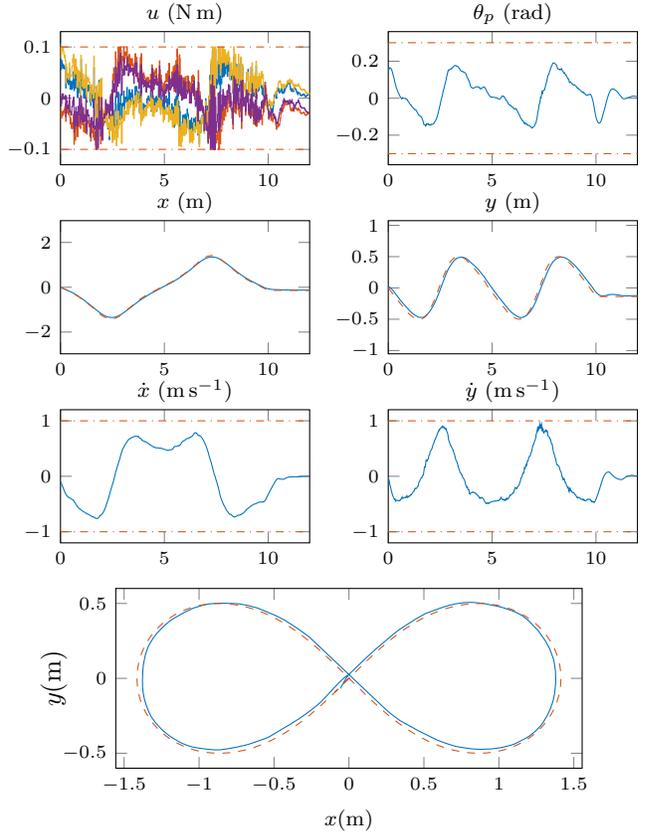


Fig. 15: Experimental trajectories for a figure-of-8 reference of 10s duration and constant ϕ .

VI. CONCLUSION

This article has demonstrated the first successful implementation of a real time constrained time-optimal infinite-horizon MPC for position and heading control of a wheeled inverted pendulum. This is demonstrated on a platform of the Collinear Mecanum Drive configuration, though through a redefinition of the nonholonomic constraints (1) and reduction of ζ this method can be adapted to the more common two-wheeled inverted pendulum.

Given the good experimental performance demonstrated by this controller on this small and highly dynamic prototype, future work would explore application of this controller to a larger system with slower dynamics. This would allow for longer optimisation execution times, allowing a larger control horizon to be used to eliminate the small tracking error present in Fig. 10 and 15. Future work will also explore introducing move blocking as discussed in Section III-H to allow for a larger n_r and higher gain feedback for the same number of decision variables, along with exploring the size and distribution of these blocks.

The development of a high level controller capable of generating sets of waypoints for this controller to follow would enable the navigation of a known environment. This would only require the selection of a small number of waypoints, located at points where a change of direction is desired. In contrast with polynomial-based trajectory generation methods

[20] this removes the requirement for the outer controller to specify a dynamically feasible and constraint satisfying trajectory between waypoints, with the dynamically feasible and time-optimal trajectory here instead derived iteratively by the MPC. The MPC demonstrated here is better suited to this task than existing MPC implementations, as by maintaining feasibility across the full reference set waypoints can be placed at arbitrarily sparse intervals, rather than needing to consider the controller's feasible reference set. The embedding of input and output constraint satisfaction into the low level controller also improves safety and robustness in the event of delay or error in a higher level controller, with the system guaranteed to safely come to rest at the end of the last specified reference.

REFERENCES

- [1] M. T. Watson, D. T. Gladwin, T. J. Prescott, and S. O. Conran, "Design and control of a novel omnidirectional dynamically balancing platform for remote inspection of confined and cluttered environments," in *2018 IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES)*. IEEE, jan 2018, pp. 473–478.
- [2] S. Reynolds-Haertle and M. Stilman, "Design and Development of a Dynamically-Balancing Holonomic Robot," Georgia Institute of Technology. Center for Robotics and Intelligent Machines, Tech. Rep., 2011.
- [3] R. P. M. Chan, K. A. Stol, and C. R. Halkyard, "Review of modelling and control of two-wheeled robots," *Annual Reviews in Control*, vol. 37, no. 1, pp. 89–103, apr 2013.
- [4] Z. Li, C. Yang, and L. Fan, *Advanced control of wheeled inverted pendulum systems*. Springer, 2013.
- [5] K. Pathak, J. Franch, and S. K. S. Agrawal, "Velocity and position control of a wheeled inverted pendulum by partial feedback linearization," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 505–513, jun 2005.
- [6] S. Kim and S. Kwon, "Nonlinear optimal control design for underactuated two-wheeled inverted pendulum mobile platform," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2803–2808, dec 2017.
- [7] N. Dini and V. J. Majd, "Model predictive control of a wheeled inverted pendulum robot," in *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*. IEEE, oct 2015, pp. 152–157.
- [8] N. Hirose, R. Tajima, N. Koyama, K. Sukigara, and M. Tanaka, "Following control approach based on model predictive control for wheeled inverted pendulum robot," *ADVANCED ROBOTICS*, vol. 30, no. 6, pp. 374–385, 2016.
- [9] M. Yue, C. An, and J.-Z. Sun, "An Efficient Model Predictive Control for Trajectory Tracking of Wheeled Inverted Pendulum Vehicles with Various Physical Constraints," *International Journal of Control, Automation and Systems*, vol. 18, no. 1, pp. 265–274, 2018.
- [10] T. Ohhira and A. Shimada, "Model Predictive Control for an Inverted-pendulum Robot with Time-varying Constraints," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 776–781, jul 2017.
- [11] J. A. Rossiter, *Model-Based Predictive Control: A Practical Approach*. Sheffield, UK: CRC Press, 2003.
- [12] J. Rossiter, "A global approach to feasibility in linear MPC," *Proc. UKACC ICC*, 2006.
- [13] S. S. Dughman and J. A. Rossiter, "Systematic and effective embedding of feedforward of target information into MPC," *International Journal of Control*, pp. 1–15, jan 2017.
- [14] —, "A survey of guaranteeing feasibility and stability in MPC during target changes," in *IFAC-PapersOnLine*, vol. 28, no. 8, 2015, pp. 813–818.
- [15] D. Simon, J. Lofberg, and T. Glad, "Reference tracking MPC using terminal set scaling," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, dec 2012, pp. 4543–4548.
- [16] B. Pluymers, J. Rossiter, J. Suykens, and B. De Moor, "The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty," in *Proceedings of the 2005, American Control Conference, 2005*. IEEE, 2005, pp. 804–809.
- [17] E. G. Gilbert and K. T. Tan, "Linear Systems with State and Control Constraints: The Theory and Application of Maximal Output Admissible Sets," *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.
- [18] G. Valencia-Palomo, M. Pelegrinis, J. A. Rossiter, and R. Gondhalekar, "A move-blocking strategy to improve tracking in predictive control," pp. 6293–6298, jun 2010.
- [19] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: a parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.
- [20] M. Shomin and R. Hollis, "Fast, dynamic trajectory planning for a dynamically stable mobile robot," in *IEEE International Conference on Intelligent Robots and Systems*, 2014, pp. 3636–3641.



Matthew T. Watson received the M.Eng. (Hons.) degree in electronic engineering from the University of Sheffield, Sheffield, U.K. in 2014. Since 2015 he has been working towards the Ph.D. degree in the Department of Electronic and Electrical Engineering at the University of Sheffield. His current research interests include embedded system development, dynamic system modelling, and real time optimal control.



Daniel T. Gladwin received the M.Eng. (Hons.) degree in electronic engineering (computer architecture) and the Ph.D. degree in automated control structure design and optimization using evolutionary computing from the University of Sheffield, Sheffield, UK, in 2004 and 2009, respectively. In 2012, he was appointed a Lecturer at the University of Sheffield and became Senior Lecturer in 2017. He is a founding member and Deputy Director of the Centre for Research into Electrical Energy Storage and Applications (CREESA) at the University of Sheffield. His current research interests include robotics, energy storage and management, optimization, and intelligent systems with over 50 publications in these areas.



Tony J. Prescott holds an MA degree in Psychology from the University of Edinburgh, U.K., an MSc degree in Applied Artificial Intelligence from the University of Aberdeen, U.K., and a PhD in Machine Learning from the University of Sheffield, UK. He is currently a Professor of Cognitive Robotics in the Department of Computer Science at the University of Sheffield. He is also a co-founder and the current Director of *Sheffield Robotics* a cross-disciplinary institute across both universities in Sheffield. He has authored over 200 publications in robotics, computational modeling, cognitive science and computational neuroscience.

Conflict of Interest Statement. Professor Prescott is a Director and Shareholder in the UK company *Consequential Robotics* which develops companion and assistive robots.



Sebastian O. Conran is CEO of Consequential Robotics and Sebastian Conran Associates, He has strong links with leading universities; is Designer in Residence at University of Sheffield's engineering faculty and visiting Professor of Innovation at Bristol Robotics Lab. He focuses on designing technologies to make them more emotionally engaging; transforming research & engineering into user experience with design thinking. He has received many awards for design & innovation; been granted over fifty patents and design patents in various fields; and holds a number of honorary doctorates and fellowships.