Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 58-18 (2024) 53–58

Tracking-in-range Formulations for Numerical Optimal Control

Nikilesh Ramesh * **Eric C. Kerrigan** ** **Yuanbo Nie** ***

* *Interdisciplinary Programmes in Engineering, University of Sheffield, S10 2TN Sheffield, UK, email: nramesh2@sheffield.ac.uk*

** *Department of Electrical & Electronic Engineering and Department of Aeronautics, Imperial College London, SW7 2AZ London, UK, email: e.kerrigan@imperial.ac.uk*

*** *Department of Automatic Control and Systems Engineering, University of Sheffield, S10 2TN Sheffield, UK, email: y.nie@sheffield.ac.uk*

Abstract: In contrast to set-point tracking which aims to reduce the tracking error between the tracker and the reference, tracking-in-range problems only focus on whether the tracker is within a given range around the reference, making it more suitable for the mission specifications of many practical applications. In this work, we present novel optimal control formulations to solve tracking-in-range problems, for problems requiring the tracker to be always in range, as well as problems allowing the tracker to go out of range, which yield overall better outcomes. Since the problem naturally involves discontinuous functions, we present alternative formulations and regularisation strategies to improve the performance of numerical solvers. The extension to in-range tracking with multiple trackers and in-range tracking in high-dimensional space are also discussed and illustrated with numerical examples, demonstrating substantial increases in mission performance in comparison to traditional set-point tracking.

Copyright © 2024 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

Keywords: Optimal Control, Nonlinear Model Predictive Control, Intelligent Vehicles

1. INTRODUCTION

The concept of tracking has been central to control engineering. While the use of set-point tracking (described as error-controlled regulation by Ashby (1956)) has become the de facto standard for such tasks, the notion of 'always keeping close to the reference' does not accurately represent the mission specifications of many practical applications. For example, when filming an automotive race event using an unmanned aerial vehicle (UAV) with finite battery resources, the goal may not necessarily be for the target car to be tracked at the centre of the UAV camera until the battery runs out, but instead to keep the car always in the view of the camera (i.e. always in range, abbreviated as 'a.i.r.'), or have the car in the view of the camera for the longest time possible, given the duration of the race (i.e. not always in range, abbreviated as 'n.a.i.r.').

The design specifications of such applications can be classified as 'in-range tracking' problems. The key characteristic of such problems is that the tracking performance of a dynamic variable is evaluated against a given range $\pm\delta$ around a time-varying reference, with 'in-range' being the desired outcome and 'out-of-range' being undesirable.

Importantly,

- Inside each sub-category of 'in-range' and 'out-of-range', there should be no preferences regarding how close the tracking is achieved against the reference point.
- The desire to stay in range should not equate to a requirement for the tracker to always stay in range. In many practical problems, going momentarily out of range may yield overall better outcomes.

In an optimisation-based control setting, the traditional optimal control problem (OCP) formulations do not represent the nature of 'in-range tracking' problems well:

- The quadratic regulation cost for set-point tracking, as in Rawlings et al. (2017), focuses on the reduction of tracking error magnitude (often subject to a trade-off with control efforts),
- For a.i.r. problems, formulating an OCP with the range specified as constraints can yield the same result as 'in-range tracking' problems. However, for solutions requiring the tracker to go momentarily out of range (i.e. n.a.i.r.), adapting the constraints as soft constraints (Kerrigan and Maciejowski, 2000) may not be sufficient, since such a formulation will still bias different solutions depending on the distance to the range boundaries.

The characteristics mentioned above are visualised in Figure 1 for a 1-D tracking problem. For the same energy consumption, in-range tracking with the soft constraint formulation achieves a 31.4 % increase in mission duration in comparison to set-point tracking by keeping the tracker always in range, but not tightly following the target. The in-range tracking formulation that allows the tracker to move out of range can further increase the mission

2405-8963 Copyright © 2024 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2024.09.009

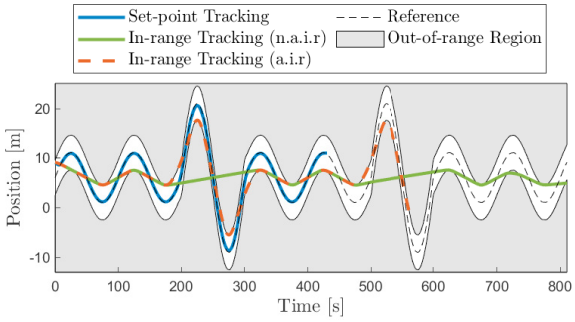


Fig. 1. Solutions to a 1-D tracking problem. All solutions have equal energy usage. 'a.i.r.' stands for always-in-range and 'n.a.i.r.' stands for not always-in-range.

duration by 245 s. With an additional in-range time of 54 s, the tracker can now track the target 43.0 % longer than set-point tracking.

In this paper, we explore suitable OCP formulations that would solve the in-range tracking problems under a trajectory optimisation setting, with the eventual aim of incorporating the design into a nonlinear model predictive control (NMPC) framework. The emphasis is on the practical implementation of the n.a.i.r. formulation, which is one of the main contributions of this work.

Sections 2 and 3 give a formal introduction to tracking problems in optimisation-based control. In Section 4, we propose techniques to improve the numerical performance of the in-range tracking problems and Section 5 extends the work to multi-agent formulations. Two examples, including a multi-tracker and a higher-dimensional tracking problems, are presented in Section 6. Finally, Section 7 highlights the concluding remarks and discussions for future work.

2. SET-POINT TRACKING WITH OPTIMAL CONTROL

A general OCP can be formulated as such

$$\min_{t_0,t_f,p,x(\cdot),u(\cdot)} \Phi(t_0,t_f,x(t_0),x(t_f),u(t_0),u(t_f),p) + \int_{t_0}^{t_f} L(x(t),u(t),p,t)\,\mathrm{d}t \tag{1a}$$

$$\text{s.t. } f(x(t),\dot{x}(t),u(t),p,t)=0, \quad \forall t\in[t_0,t_f] \tag{1b}$$

$$g(x(t),\dot{x}(t),u(t),\dot{u}(t),p,t)\le 0, \quad \forall t\in[t_0,t_f] \tag{1c}$$

$$\phi(t_0,t_f,x(t_0),x(t_f),u(t_0),u(t_f),p)\le 0, \tag{1d}$$

with Φ the Mayer cost, L the Lagrange cost, f the dynamics constraint functions related to the system differential-algebraic equations, g the inequality path constraint functions and ϕ the boundary constraint functions. This is an infinite-dimensional optimisation problem with decision variables $x:\mathbb{R}\to\mathbb{R}^n$ (the states) and $u:\mathbb{R}\to\mathbb{R}^m$ (the inputs) being functions or trajectories. Static decision variables p can be included as part of the problem formulation as well, and the start time t_0 and end time t_f of the problem can also be free. The objective functional (1a) is often denoted by a single variable J with optimal value denoted as J^*.

2.1 Set-point tracking

In optimisation-based control, set-point tracking is typically achieved through the minimisation of the following quadratic regulation cost

$$J=\int_{t_0}^{t_f} e(t)^\top Q e(t)+u(t)^\top R u(t)\,\mathrm{d}t, \tag{2}$$

with $e(t):=x(t)-x_r(t)$ the instantaneous tracking error between x and reference x_r. $Q\succeq 0\in\mathbb{R}^{n\times n}$ and $R\succ 0\in\mathbb{R}^{m\times m}$ are weighting matrices addressing the relative trade-offs between different variables and different terms in the set-point tracking.

The quadratic regulation cost is typically used under a fixed horizon. As the tracking errors are integrated along the time horizon, having a free t_f would naturally favour solutions with short horizons. To solve problems that require the tracking duration to be maximised, the common approaches are to

- iteratively increases t_f, which is fixed in the OCP formulation, until the OCP becomes infeasible or the tracking performance is no longer acceptable, or
- add additional trade-off terms in the objective formulation, e.g. $-\omega t_f$ with ω a trade-off weight.

In practice, the idea of having an acceptable range around the reference also often applies to set-point tracking, to evaluate the tracking performance posteriorly or to be implemented as state or output constraints to ensure the tracking performance. Nevertheless, the preference to reduce tracking error always exists in such formulations.

3. IN-RANGE TRACKING

In contrast, in-range tracking only focuses on whether the tracker is within a given range $\pm\delta$ around a reference. Within each outcome category of 'in-range' and 'out-of-range', in-range tracking is invariant to the exact deviation from the reference. Depending on whether 'out-of-range' instances are allowed, in-range tracking problems can be tackled differently.

3.1 Always-in-range

The always-in-range (a.i.r.) formulation leverages the inequality path constraint (1c) to achieve tracking-in-range. This approach strictly confines the trajectory to the desired tracking range $\pm\delta$. We can formulate the constraint as such for a single tracker:

$$(x(t)-x_r(t))^2-\delta^2\le 0, \quad \forall t\in[t_0,t_f]. \tag{3}$$

The a.i.r. problems are typically formulated with minimum energy or control effort objectives using fixed t_f, or with a time duration maximisation objective using free t_f. When such path constraints are specified as hard constraints, the state must be initialised within the range and stay in the range. Since this may not always be feasible in a real-world engineering situation, practical implementation will typically see (3) implemented as soft constraints by augmenting the original objective J with additional terms penalising constraint violations, introducing a trade-off between the objective minimisation and constraint satisfaction.

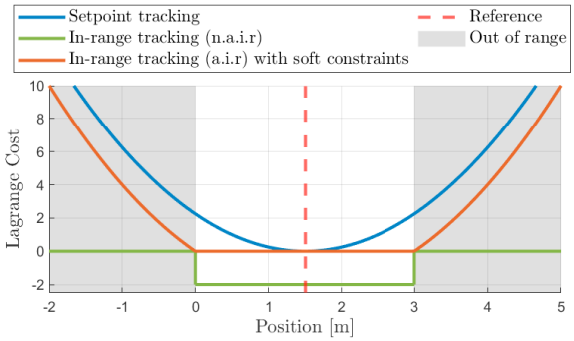


Fig. 2. Lagrange cost comparison between different formulations with $x_r = 1.5$ m and $\delta = 1.5$ m

3.2 Not always-in-range

In not always-in-range (n.a.i.r.) problems, in-range tracking is achieved via mathematically encoding this tracking behaviour in the objective function along with the original objectives. The stage cost formulation should yield penalties when the tracked variables fall outside the range and yield rewards otherwise, with no preference in the magnitude of deviation from the reference in both cases.

A straightforward choice is to use an indicator function for the Lagrange cost in the form

$$L_H := \begin{cases} \alpha & \text{for } x_r - \delta \leq x \leq x_r + \delta \\ \beta & \text{otherwise} \end{cases} \tag{4}$$

with α and β constants representing respectively the reward for the tracker to be in the range of the target, and the penalty for staying outside the range. A graphical representation of such a function with $\alpha = -2$ and $\beta = 0$ is shown in Figure 2, where a range radius $\delta = 1.5$ m is considered for a target located at a position $x_r = 1.5$ m. The differences between the Lagrange cost for setpoint tracking and the soft constraint formulation for a.i.r. tracking can also be observed in the figure.

The desire for the tracker to stay in range can be achieved through the minimisation of the Lagrange cost, i.e.

$$\min_{t_0, t_f, p, x(\cdot), u(\cdot)} \int_{t_0}^{t_f} L_H(x(t), u(t), p, t)\, \mathrm{d}t \tag{5}$$

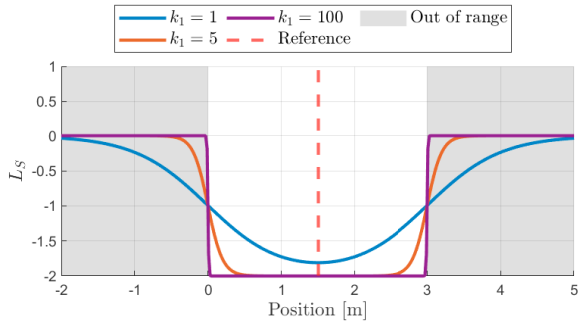
subject to relevant constraints (1b)–(1d). If the problem specifications have other original objectives, e.g. to minimise energy consumption or control efforts, they need to be augmented to (5) with suitable trade-off weights.

4. IMPROVING THE NUMERICAL PERFORMANCE OF IN-RANGE TRACKING PROBLEMS

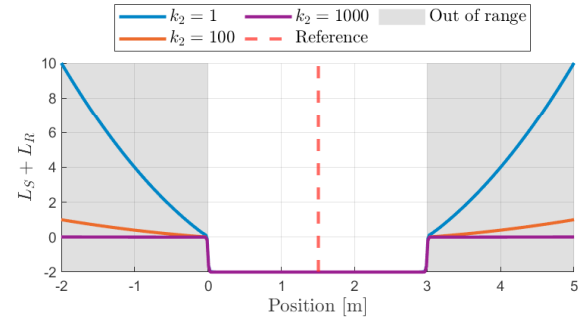
4.1 Choice of constants in the indicator function

Due to the integration in (1a) for the Lagrange cost, the choice of α and β will not be shift-invariant, with some design choices better suited for numerical computations than others. Our key observations are

- Having $\alpha = 0$ and $\beta \geq 0$ should be avoided. This is because different solutions with the tracker staying in range throughout the time horizons will all have zero costs, regardless of the duration of the problem $[t_0, t_f]$ thus resulting in non-uniqueness of solutions.



(a) Cost with varying k_1



(b) Regularised cost with varying k_2 and constant $k_1 = 100$

Fig. 3. Effect of constants k_1 and k_2 on the stage cost approximation

- For problem specifications solely focusing on maximising the in-range time, having $\alpha < 0$ and $\beta = 0$ is often a good choice, since the duration for which the tracker stays outside the range is not directly penalised. This would allow the tracker to comfortably stay outside the range if it is beneficial to increase the overall in-range time.

4.2 Smooth approximation of the indicator function

The design of the Lagrange cost as a discontinuous indicator function will pose many challenges when using derivative-based numerical solvers. Therefore it is desirable to use designs of continuous-differentiable expressions to approximate the desired shape of the original function.

The indicator function (4) can be viewed as the combination of two Heaviside functions (Abramowitz and Stegun, 1968), for which different smooth approximations exist in the literature, e.g. by Iliev et al. (2015).

As an example, with $\alpha = -2$ and $\beta = 0$,

$$\begin{aligned} L_H \approx L_S := &\tanh\left(k_1((x - x_r) - \delta)\right) \\ &+ \tanh\left(k_1(-(x - x_r) - \delta)\right) \\ = &\frac{2}{1 + e^{-2k_1((x-x_r)-\delta)}} \\ &+ \frac{2}{1 + e^{-2k_1(-(x-x_r)-\delta)}} - 2. \end{aligned} \tag{6}$$

With k_1 a real-valued parameter. Figure 3a illustrates the effect of choosing k_1 on the smooth approximation of L_H.

4.3 Regularisation

Despite the smooth approximation, the numerical solution of the in-range tracking OCP could still be challenging.

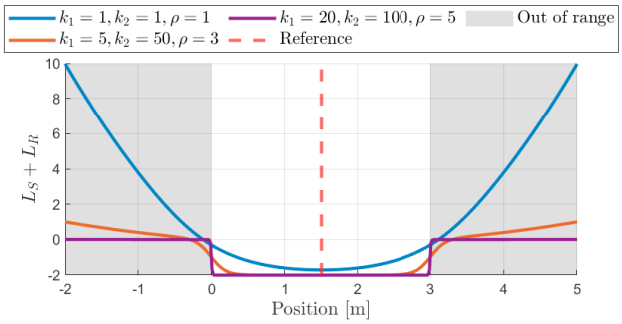


Fig. 4. Changes of the Lagrange cost as k_1, k_2 and ρ increase iteratively

We find it practically beneficial to add an additional regularisation term

$$L_R := \frac{1}{k_2}(x - x_r)^2, \tag{7}$$

with regularisation weight $k_2 \to \infty$ for $L_R \to 0$.

To further reduce the impact of the regularisation term on the trajectory of the tracker when it is in range, we can optionally apply the quadratic regularisation only for the out-of-range regions by having

$$L_R := \frac{1}{k_2}\max\{(x - x_r)^2 - \delta^2, 0\}, \tag{8}$$

and employ a smooth approximation of the maximum operator for the computation of the numerical value. For example, Kreisselmeier and Steinhauser (1979) proposed

$$\max\{s_1, s_2, \ldots s_n\} \lessapprox \frac{1}{\rho} \cdot \log(e^{\rho s_1} + e^{\rho s_2} + \ldots e^{\rho s_n}), \tag{9}$$

with the accuracy of the approximation improving as $\rho \to \infty$. Figure 3b illustrates the effect of varying k_2 with k_1 and ρ fixed.

Alternatively, we can define

$$L_R := \frac{L_I}{k_2}\left((x - x_r)^2 - \delta^2\right), \tag{10}$$

with L_I the smooth approximation of an indicator function

$$L_I \approx \begin{cases} 0 & \text{for } x_r - \delta \le x \le x_r + \delta \\ 1 & \text{otherwise} \end{cases} \tag{11}$$

using the same method as in Section 4.2.

4.4 Solution strategy

With the smooth approximations and the regularisation, the desire for the tracker to stay in-range can be achieved by solving the following OCP

$$\min_{t_0, t_f, p, x(\cdot), u(\cdot)} \int_{t_0}^{t_f} L_S + L_R \, \mathrm{d}t \tag{12}$$

subject to relevant constraints (1b)–(1d).

In practice, it is unlikely that a single choice of k_1, k_2 and ρ would be sufficient for the solver to efficiently and reliably converge to a solution. Therefore, the OCP may need to be iteratively solved with warm-starting, with k_1, k_2 and ρ increasing after each iteration. Figure 4 illustrates the convergence of the Lagrange cost iteratively towards the original cost (4), with $\alpha = -2$ and $\beta = 0$.

5. IN-RANGE TRACKING WITH MULTIPLE TRACKERS

The advantages of in-range tracking are greater in scenarios with multiple trackers, to collaboratively track the reference. The corresponding tracking requirement is typically specified as the desire to have at least one tracker in the range of the target.

In terms of the OCP formulation, for a.i.r. problems, the path constraints (3) would be reformulated as

$$\min\{(x_i - x_r)^2\}_{i=1}^N - \delta^2 \le 0. \tag{13}$$

for a problem with N agents with x_i the position of the ith agent. For n.a.i.r. problems, the objective will be based on the minimum value of the Lagrange cost across all N agents, i.e.

$$L_S + L_R := \min\{L_{S_i} + L_{R_i}\}_{i=1}^N. \tag{14}$$

The minimum operator can be non-smooth and non-differentiable, posing challenges that need to be resolved for practical computation benefits. Here we present two alternatives levering again the smooth maximum operator or using the complementary conditions. Here we will only demonstrate the n.a.i.r. problems for simplicity, as the treatment for a.i.r. problems is analogous.

5.1 Using the smooth maximum operator

As the first step, the minimum operator can be reformulated using the maximum operator as follows

$$L_S + L_R := -\max\{-(L_{S_i} + L_{R_i})\}_{i=1}^N. \tag{15}$$

Due to particular characteristics of the smooth maximum approximation, direct use of (9) in (15) could lead to undesirable behaviours. For a given ρ, the accuracy of the smooth maximum would decrease when the values of all elements are similar. Added to the fact that the approximation is always larger than the true maximum, the direct use of the smooth maximum would yield a lower value for the Lagrange cost when all trackers are in range. Therefore, such an implementation would unintentionally introduce biases in the solution, in contradiction to the original specification.

Practical workarounds are available to mitigate the impact. Problems with choices of $\alpha < 0$ and $\beta = 0$ in (4), for instance, can compute

$$L_S + L_R := \max\{-\gamma \max\{-(L_{S_i} + L_{R_i})\}_{i=1}^N, \alpha\} \tag{16}$$

instead, using the smooth maximum approximation with $\gamma > 1$ a sufficiently large positive constant that can be flexibly chosen. To demonstrate using an example, consider a two-agent tracking problem with $\alpha = -2$ and $\beta = 0$. With at least 1 tracker in the range, (15) should yield a value of -2 but the use of a smooth maximum approximation with $\rho = 1$ would yield -2.69 for the instances when both trackers are in range, and yield -2.13 when only 1 tracker is in range. When using (16) instead with $\gamma = 6$, the Lagrange cost with the smooth maximum approximation improves to -2.00 in both cases.

5.2 Using complementary conditions

In optimisation literature, the computation of the minimum value operator can be avoided by reformulating the

problem into a mathematical program with complementarity conditions (MPCC). Betts (2010) presented the MPCC reformulation for the minimum value operator which can be adapted to the multi-agent in-range problem.

For a two-agent problem, if we let

$$\begin{aligned} l_1 &= (L_{S_1} + L_{R_1}) \\ l_2 &= (L_{S_2} + L_{R_2}) \end{aligned} \tag{17}$$

then

$$\min\{L_{S_i} + L_{R_i}\}_{i=1}^{2} = l_1 + (l_1 - l_2)q, \tag{18}$$

if q is chosen to minimise $(l_1 - l_2)q$ subject to $-1 \leq q \leq 0$. This problem can be solved directly as a bi-level optimisation problem or reformulated as an MPCC by obtaining the necessary conditions for optimality for the inner optimisation problem and implementing them as constraints in the form of complementarity conditions for the "outer" optimisation. In this case, we would have

$$\begin{aligned} 0 \leq \lambda_1 \perp (1+q) \geq 0, \\ 0 \leq \lambda_2 \perp -q \geq 0, \end{aligned}$$

with $\perp$ representing the complementarity relationship and λ_1 the Lagrange multiplier for inequality constraint $-1 \leq q$ and λ_2 the Lagrange multiplier for $q \leq 0$.

Although the use of complementary conditions leads to a more accurate representation of the minimum operator, this framework is more difficult to scale with an increasing number of agents, both in terms of the complexity of the problem formulation and the number of decision variables. In addition, solving nonlinear programming problems arising from MPCC can be challenging due to the violation of constraint qualifications (Betts, 2010). As a result, our experience shows that the smooth maximum is a practically more attractive solution for multiple tracker problems.

6. NUMERICAL EXAMPLES

To demonstrate the advantages of in-range tracking, two additional example problems are presented to focus on different aspects. All problems are transcribed using the toolbox `ICLOCS2` (Nie et al., 2018), and solved with interior point NLP solver `IPOPT` (Wächter and Biegler, 2006) to a relative convergence tolerance of 10^{-9}.

6.1 Multi-tracker 1D in-range tracking with out-of-range possibilities (n.a.i.r.)

The first example is a 1D in-range tracking problem with two trackers, with the possibility for the tracker to stay out-of-range if desirable. The system dynamics is a double integrator representing simplified UAV dynamics

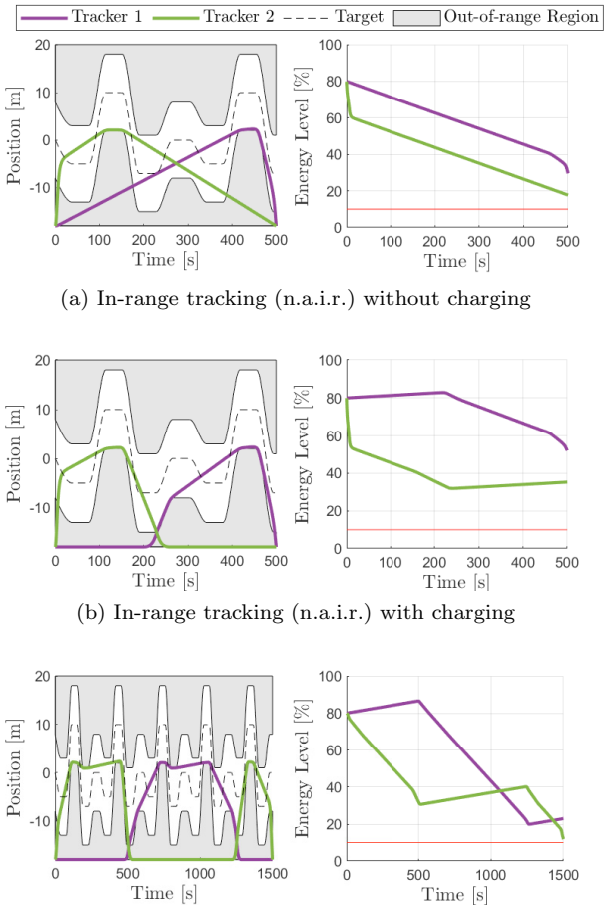
$$\ddot{x}_i = \frac{u_i(t)}{10}, \text{ for } i = 1, 2 \tag{19}$$

together with a simplified energy consumption model with the rate of discharge of the battery state-of-charge (E_i in %) modelled to have terms dependent on the tracker's velocity v_i and the input u_i, and a constant term to account for a pseudo-hovering state:

$$\dot{E}_i = -0.085 - (0.283u_i)^2 - 0.566v_i^2, \text{ for } i = 1, 2. \tag{20}$$

In this example, we will also explore the changes in the solution by using

$$\begin{aligned} \dot{E}_i = &- 0.05(0.7 + \tanh(x_i - x_b)) \\ &- (0.283u_i)^2 - 0.566v_i^2, \end{aligned} \tag{21}$$



(a) In-range tracking (n.a.i.r.) without charging

(b) In-range tracking (n.a.i.r.) with charging

(c) Long horizon in-range tracking (n.a.i.r.) with charging

Fig. 5. Solutions to multi-tracker 1D in-range tracking problems

instead. This dynamics equation includes a slow wireless charging capability when the vehicles are near their base station x_b.

The main objective is to maximise the in-range tracking time for fixed t_f with (12), with a secondary objective to minimise the total energy consumption through the maximisation of $E_1(t_f) + E_2(t_f)$. The initial state-of-charge for the batteries, $E_1(t_0)$ and $E_2(t_0)$, are both fixed at 80%. The trackers are initialised at the base and are required to come back to the base station at $x_b = -18\,\text{m}$ at the end of the mission with at least 10% of battery remaining.

The problems are solved using the automatic regularisation feature of `ICLOCS2` with $k_1 = [1, 10, 40]$ and $k_2 = [10^5, 3 \times 10^5, 10^6]$ respectively for each iteration, and with a choice of $\rho = 1$. The solution trajectories are illustrated in Figure 5.

Comparing Figure 5a and 5b, it can be observed that the opportunity to charge at the base has led to substantial changes in the solution: the duration for which both trackers are in-range at the same time has been significantly reduced. With charging and in-range tracking working together, Figure 5c shows that the two trackers can easily keep the target in range for more than 1500 seconds, whereas standard set-point tracking for this prob-

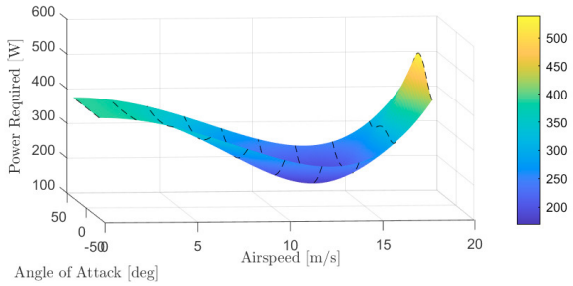


Fig. 6. Power required for steady level flight as a function of airspeed and angle of attack

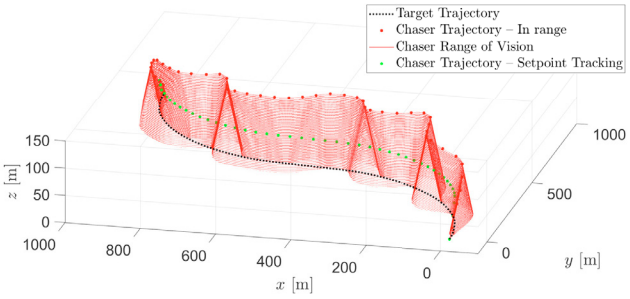


Fig. 7. Solution to a 3D always-in-range problem formulation with fixed wing dynamics

lem would typically only allow an endurance of around 350 seconds for each tracker.

6.2 Single-tracker always-in-range tracking (a.i.r.) in 3D

The simplified model in the previous example has certain characteristics that may give the misconception that in-range tracking leads to slow movements of the tracker. We aim to use this second example of 3D in-range tracking with fixed-wing UAV to demonstrate an opposite case.

Based on Liu et al. (2017), the dynamics for the fixed-wing UAV have different characteristics and the power required to maintain steady flight is no longer the lowest at zero airspeed, as shown in Figure 6. In fact, there will be an optimal airspeed to achieve maximum endurance.

We extended the a.i.r. formulation (3) to a 3D problem, effectively representing an UAV camera having zero-vision at a height of 0 m, and a 40 m radius vision when flying at the ceiling of 130 m. The aim is to keep the target always in the range of the vision. This results in an interesting solution presented in Figure 7. Distinct from the previous example, fixed-wing aerodynamics give rise to an optimal trajectory where loitering and covering more distance is more efficient. The in-range formulation leverages the optimal cruise conditions to conserve energy, as opposed to set-point tracking which consumes 6.1% more energy.

7. CONCLUSIONS AND FUTURE WORK

A framework for in-range trajectory optimisation has been presented in this paper. The in-range formulation addresses a number of shortcomings of traditional set-point tracking approaches for some real-world applications. By being invariant to the exact deviation from the reference, the optimal solutions obtained can have improved endurance or reduced energy consumption, as shown in the examples. Furthermore, multiple mathematical techniques to improve the performance of numerical solutions to these problems, such as smooth approximations of indicator and maximum operator functions, have been discussed.

Future work would focus on numerically efficient formulations to extend the not-always-in-range problem to higher dimensions, and a detailed analysis of the framework in terms of mathematical properties. Further work will also investigate the solution of in-range tracking problems in closed-loop.

ACKNOWLEDGEMENTS

The work of Alvaro Pons Pelufo, Ying Ju and Giliano-Nicolae Bazili on this topic during their Master's projects at Imperial College London is gratefully acknowledged.

REFERENCES

Abramowitz, M. and Stegun, I.A. (1968). *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office.

Ashby, W.R. (1956). *An introduction to cybernetics*. Chapman & Hall Ltd.

Betts, J.T. (2010). *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, Second Edition*. Society for Industrial and Applied Mathematics, second edition. doi:10.1137/1.9780898718577.

Iliev, A.I., Kyurkchiev, N., and Markov, S. (2015). On the approximation of the cut and step functions by logistic and gompertz functions. *Biomath*, 4(2), ID–1510101.

Kerrigan, E.C. and Maciejowski, J.M. (2000). Soft constraints and exact penalty functions in model predictive control. *Proc. UKACC International Conference (Control 2000)*.

Kreisselmeier, G. and Steinhauser, R. (1979). Systematic control design by optimizing a vector performance index. *IFAC Proceedings Volumes*, 12(7), 113–117. doi: https://doi.org/10.1016/S1474-6670(17)65584-8. IFAC Symposium on computer Aided Design of Control Systems, Zurich, Switzerland, 29-31 August.

Liu, Z., Sengupta, R., and Kurzhanskiy, A. (2017). A power consumption model for multi-rotor small unmanned aircraft systems. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 310–315. doi:10.1109/ICUAS.2017.7991310.

Nie, Y., Faqir, O., and Kerrigan, E.C. (2018). ICLOCS2: Try this optimal control problem solver before you try the rest. In *United Kingdom Automatic Control Council (UKACC) International Conference on Control*, 336–336. IEEE.

Rawlings, J.B., Mayne, D.Q., and Diehl, M. (2017). *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57.