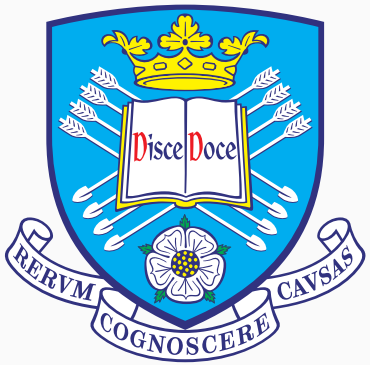
Deep Reinforcement Learning for Process Control in Powder Bed Fusion

Supervisor

George Panoutsos

Candidate

Ricardo Dominguez-Olmedo



Department of Automatic Control & Systems Engineering
The University of Sheffield

15th May 2019

Table of contents

Background

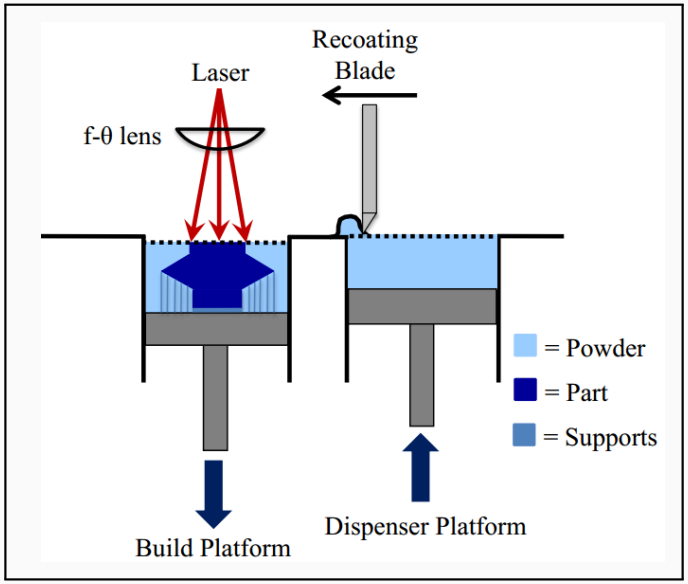
Deep reinforcement learning (DRL)

- Control learned autonomously through interaction
  - Observe consequence of actions and adapt behaviour
- No system identification or controller design required
  - Time consuming
  - Application-specific knowledge necessary
- Has only been used in robotic systems

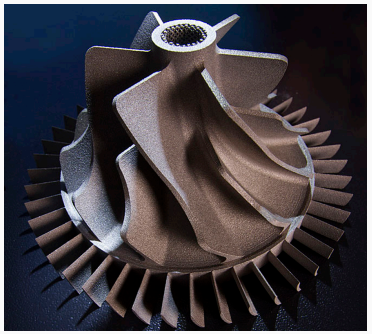
Robotic grasping task

Powder bed fusion (PBF)

- Additive manufacturing technology
  - 3D objects build by stacking layers of material
  - Very high geometric freedom
- Difficult to reliably manufacture high-quality products
  - "Optimal" parameters found by trial-and-error
  - Closed-loop control necessary to improve quality (surface temperature?)



Powder bed fusion process



Radial compressor

Project aim

Use DRL for thermal control of a PBF process.

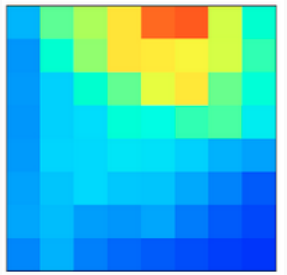
- Assess suitability of DRL for industrial process control
- PBF is very challenging...
  - Highly non-linear
  - High-dimensional
  - No known model
  - Expensive

Problem formulation

Manufacturing system

- Controllable parameters: laser power and scan speed
- Pyrometer to measure surface temperature
  - Heat map of the entire layer surface
  - Divide into "cells" $\rightarrow$ system state
- **Objective:** maintain an optimal surface temperature

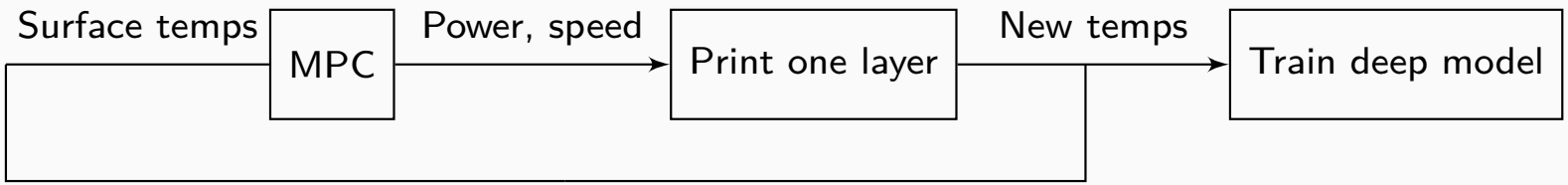
The AconityMINI machine



Surface heat map
(red is hottest)

The PETS Algorithm

- Iteratively train a model of the dynamics using data collected so far
  - Model is a deep neural network
- Use Model Predictive Control (MPC) over learned model
  - Control inputs over plan horizon optimised using a black-box optimiser
  - Cost function: exponential negative quadratic error $-e^{-k(x-r)^T(x-r)}$



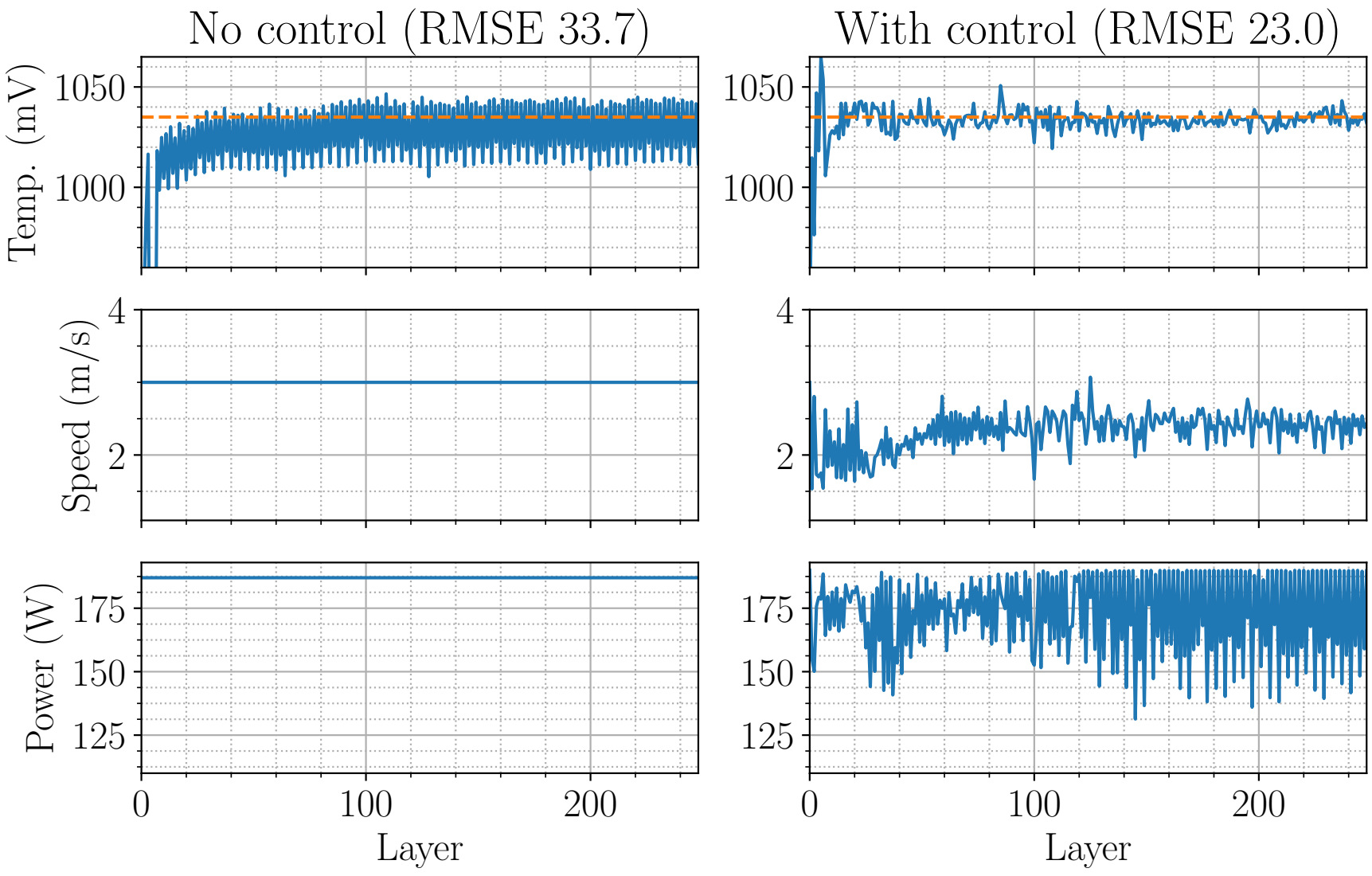
Experiments

- Two different materials:
  - CM247, 10x10x5mm cuboids
  - Haynes 282, 5x5x5mm cuboids
- 20 cuboids built per experiment
  - 10 with constant optimal parameters
  - 10 with thermal control

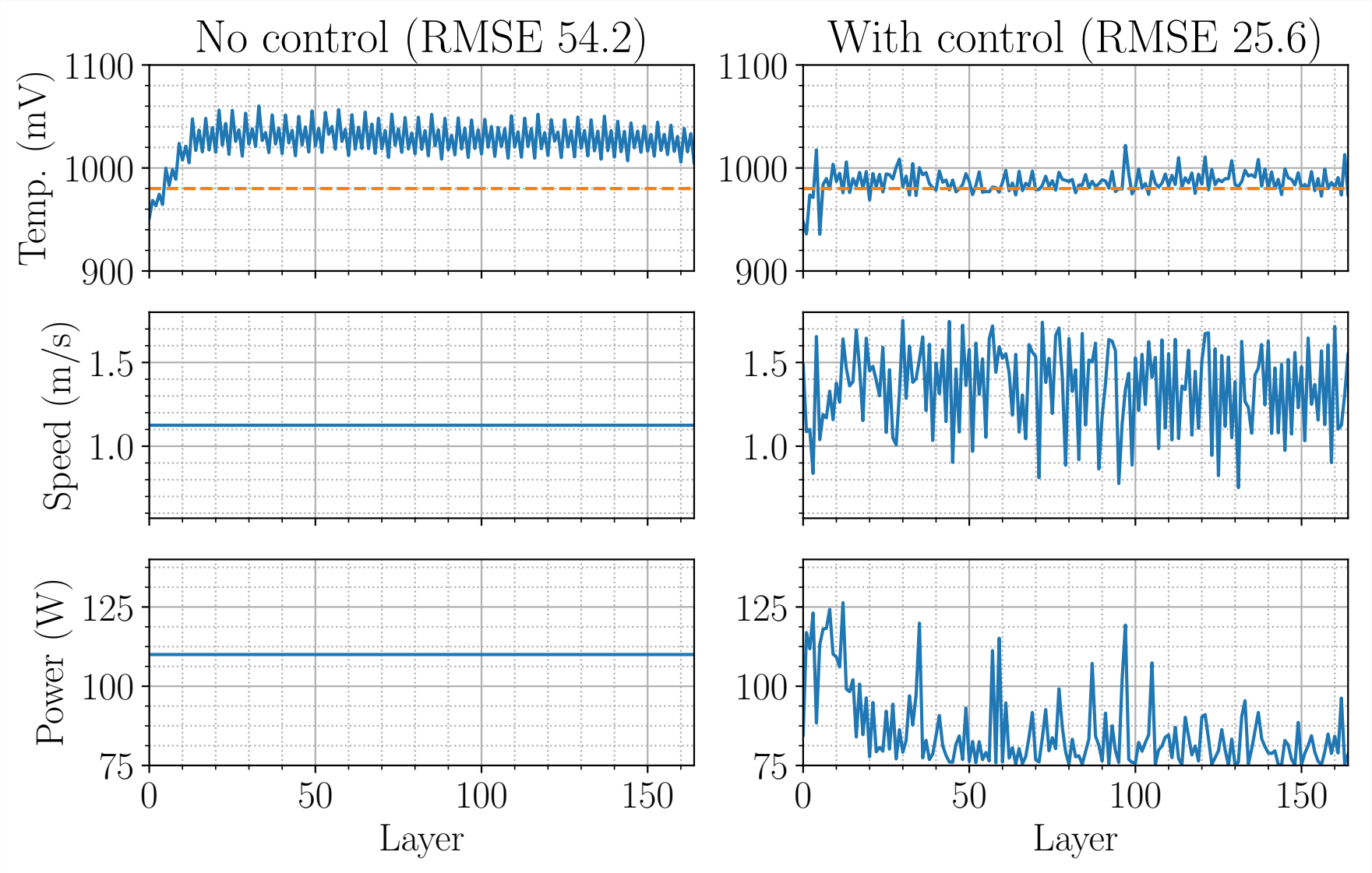
Sample cuboid

Results

Experiment I: CM247

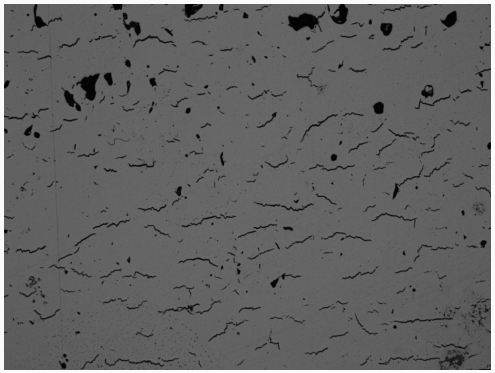


Experiment II: Haynes 282

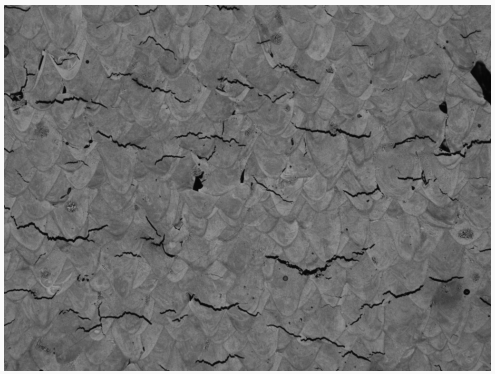


Build quality

CM247



No control



With control

Haynes 282

Density of the samples built.

	Density (%)	
	Mean	Minimum
No control	**97.9±0.6**	97.2
With control	95.9±0.7	95.4

Why does it not improve?

- Relationship temperature - quality not well understood
- Drastic changes of input parameters

Conclusions

Advantages of our DRL approach in PBF

- Determining optimal parameters via trial-and-error is expensive
  - Tens of thousands of pounds and days
  - Redone for every new object manufactured
- DRL leads to good quality at much lower cost
- Quality - Cost trade-off very appealing in some instances

Novel contributions

- 1st study to use DRL for process control of a real-world system
  - Very strong control performance for a complex process
  - No system identification or controller tuning required
- 1st study to demonstrate closed-loop thermal control in PBF
  - Research: further study relationship thermal signatures - build quality
  - Industry: acceptable product quality without expensive parameter tuning

Questions?

PETS Algorithm: General

1: Initialise dataset $\mathbb{D}$ (e.g. data from a single trial using a random control inputs)
2: **for** Trial $k = 1$ to K **do**
3: Train dynamics model $\widetilde{f}$ given $\mathbb{D}$.
4: **for** Time $t = 0$ to task time-horizon H **do**
5: Obtain optimal control inputs $u^*_{t:t+T}$ using CEM and TS
6: Execute first control input u^*_t
7: Store outcome: $\mathbb{D} \leftarrow \mathbb{D} \cup \{x_t, u^*_t, x_{t+1}\}$

$$\widetilde{f}(x_{t+1} \mid x_t, u_t) = Pr(x_{t+1} \mid x_t, u_t) = \mathcal{N}(\mu(x_t, u_t), \sigma(x_t, u_t)) \tag{1}$$

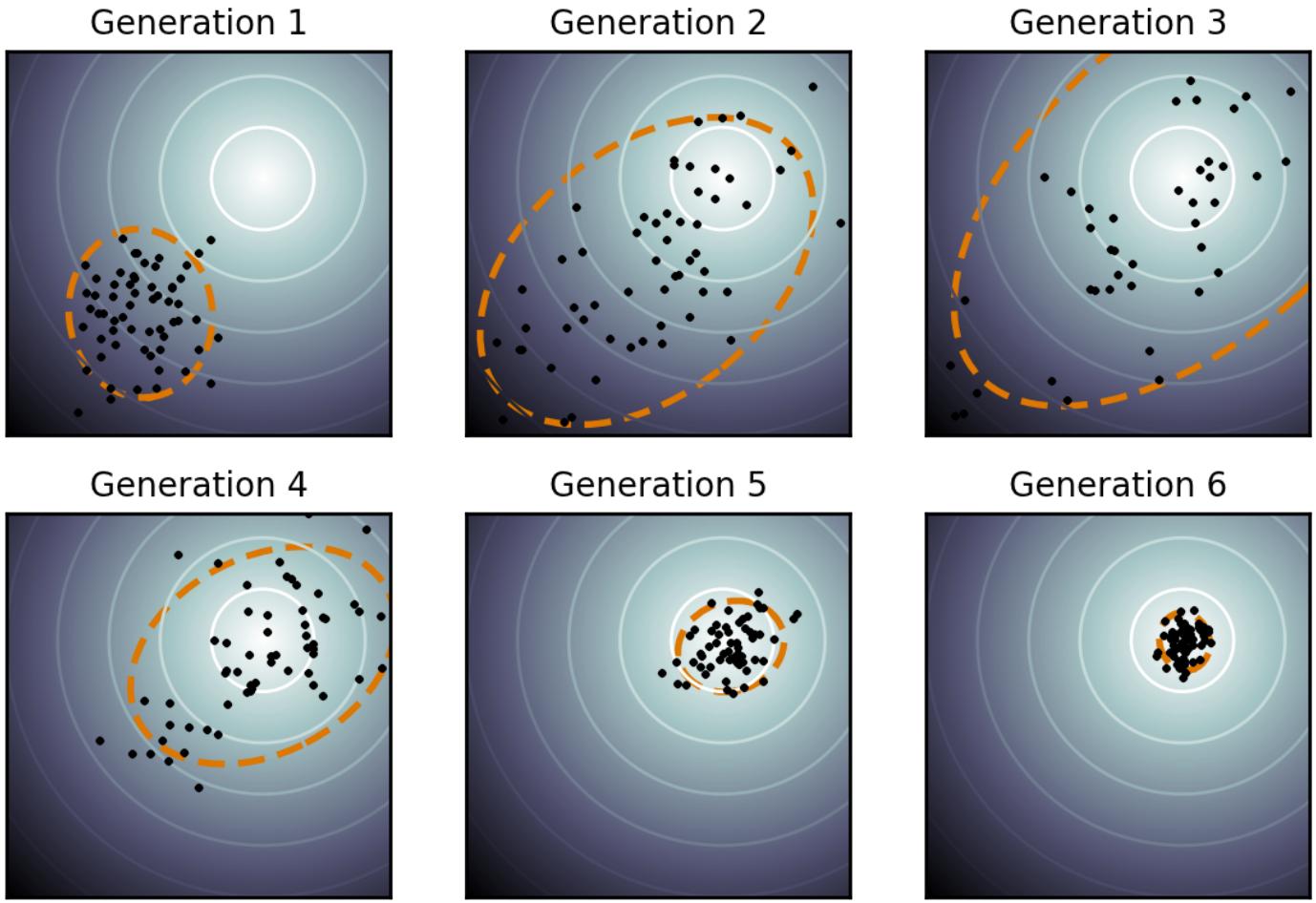
PETS Algorithm: Trajectory Sampling

for Particle $p = 1$ to P **do**
    Initialise particle to current system state $x_t^p \leftarrow x_t$
    Uniformly sample a bootstrap $b \leftarrow \{1, \ldots, B\}$
    for Time-step $\tau = t$ to $t + T$ **do**
        Sample next system state $x_{\tau+1}^p \leftarrow \widetilde{f}_{\theta_b}(x_\tau^p, u_\tau)$
        if TS variant is *TS1* **then**
            Uniformly resample a bootstrap $b \leftarrow \{1, \ldots, B\}$
return Expected cost $J \leftarrow -\frac{1}{P} \sum_{p=0}^{P} \sum_{\tau=t}^{t+T} \mathcal{R}(x_{\tau+1}^p, u_\tau)$

PETS Algorithm: Optimisation

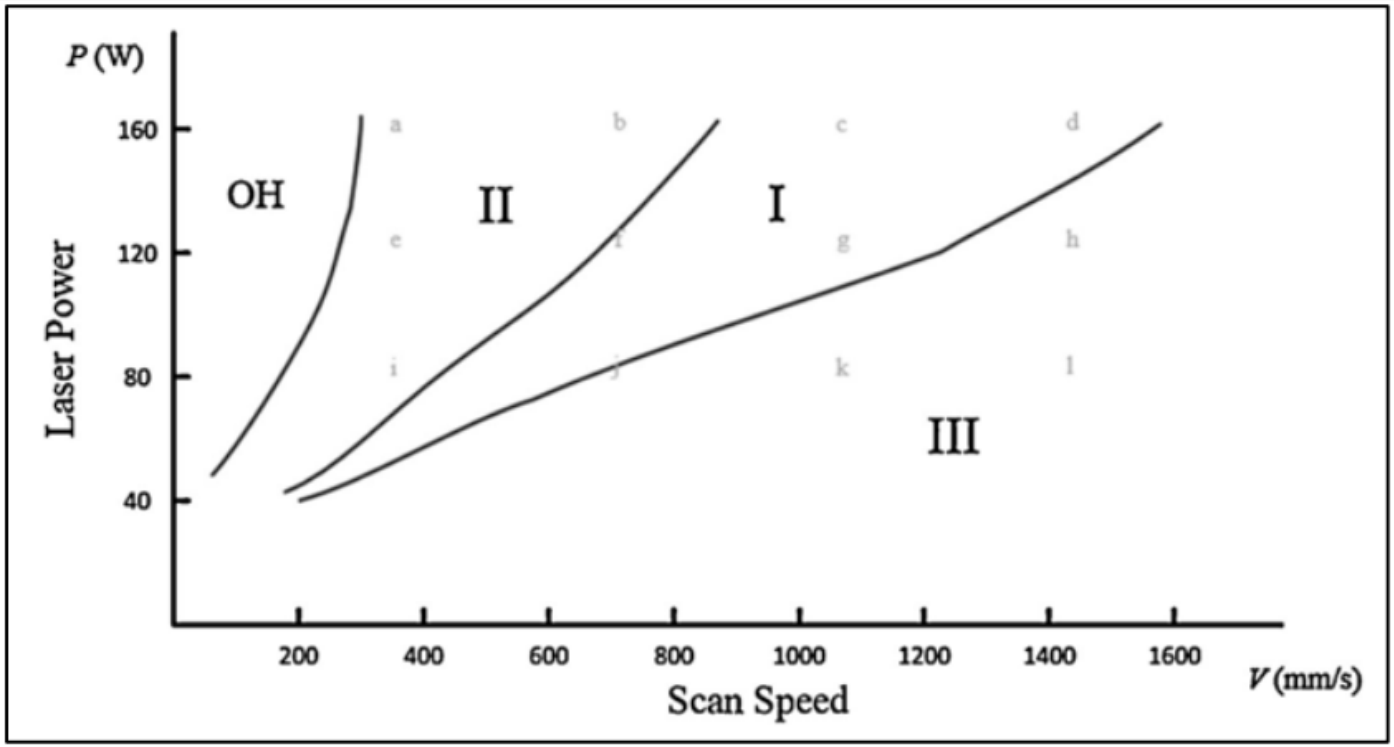
for $i = 1$ to MaxIters **do**
  Sample N control input sequences $u^1_{t:t+T}, \ldots, u^N_{t:t+T} \sim \mathcal{N}(\mu_{u_{t:t+T}}, \sigma^2_{u_{t:t+T}})$
  Evaluate expected cost $J_1, \ldots, J_N$ of $u^1_{t:t+T}, \ldots, u^N_{t:t+T}$ using TS
  Pick the N_{elites} control input sequences $u'^1_{t:t+T}, \ldots, u'^{N_{\text{elites}}}_{t:t+T}$ with lowest cost
  Compute mean $\mu'_{u_{t:t+T}}$ and variance $\sigma'^2_{u_{t:t+T}}$ of elites $u'^1_{t:t+T}, \ldots, u'^{N_{\text{elites}}}_{t:t+T}$.
  Update mean $\mu_{u_{t:t+T}} \leftarrow \alpha\mu_{u_{t:t+T}} + (1-\alpha)\mu'_{u_{t:t+T}}$
  Update variance $\sigma^2_{u_{t:t+T}} \leftarrow \alpha\sigma^2_{u_{t:t+T}} + (1-\alpha)\sigma'^2_{u_{t:t+T}}$
  Constrain variance $\sigma^2_{u_{t:t+T}} \leftarrow \min(\sigma^2_{u_{t:t+T}}, \frac{(\mu_{u_{t:t+T}} - u_{\min})^2}{4}, \frac{(u_{\max} - \mu_{u_{t:t+T}})^2}{4})$
  if Convergence criteria is met: $\max(\sigma^2_{u_{t:t+T}}) \leq \epsilon$ **then**
    break
return Optimal control input sequence $u^*_{t:t+T} \leftarrow \mu_{u_{t:t+T}}$

CEM optimisation



CEM Optimisation

Optimal energy envelopes in PBF



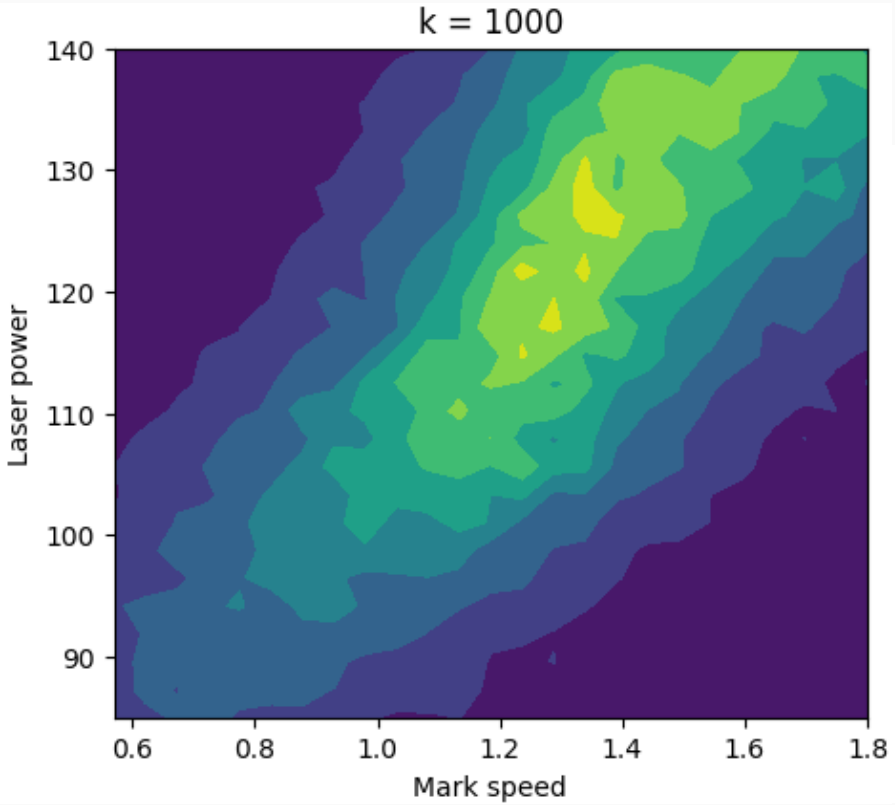
Example of energy density envelopes [**?**].

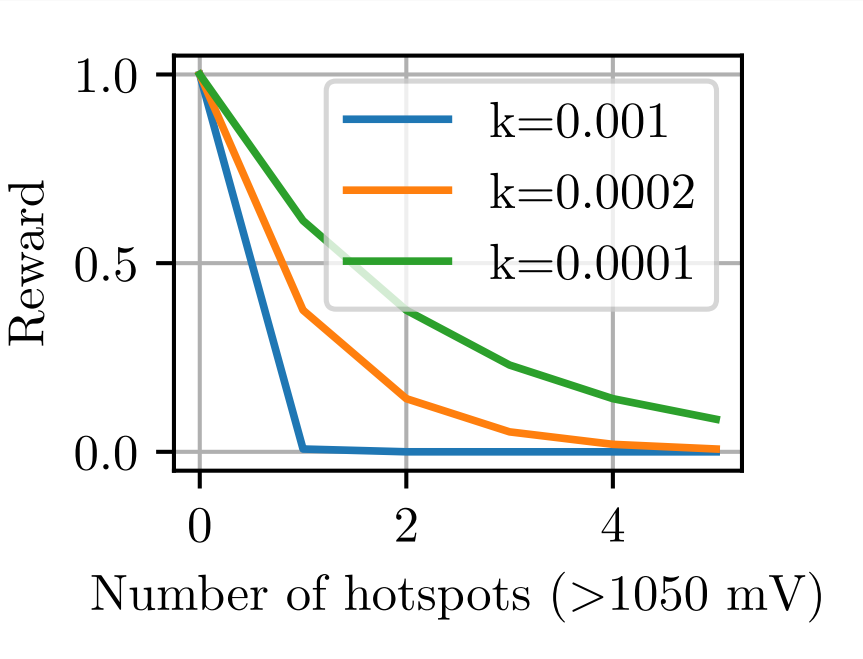
Choice of number of cells

Model accuracy and number of cells

Number of cells	MSE
4	389±12
16	**274±11**
64	293±7

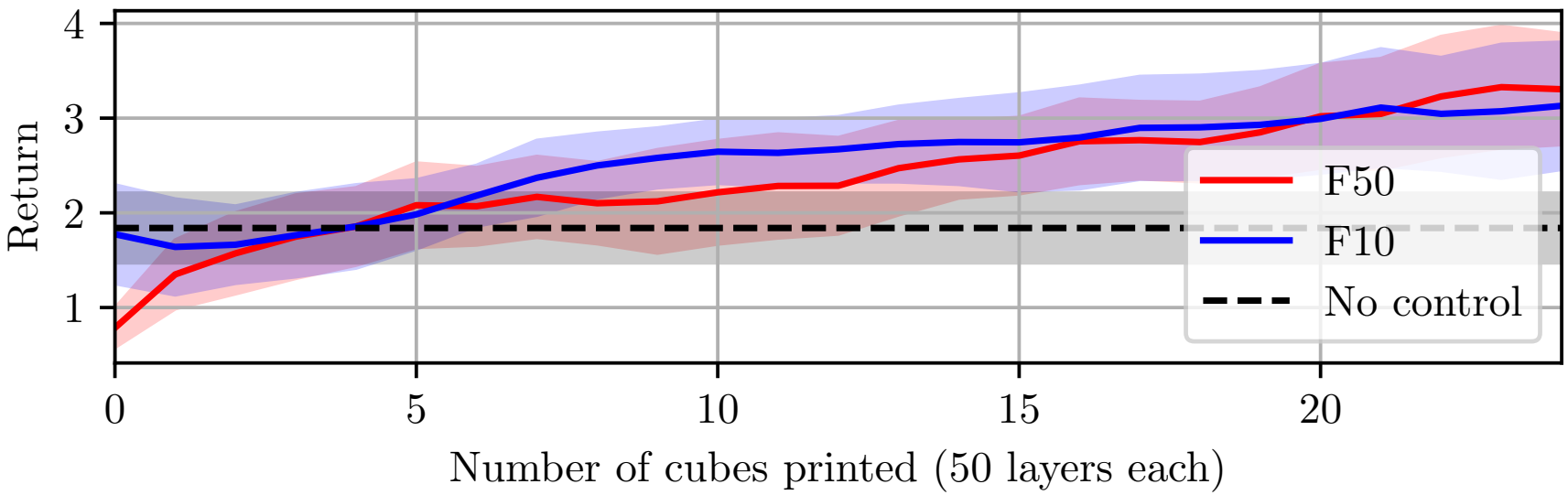
Choice of k





(a) Energy density envelopes for $k = 0.001$. Darker colours indicate lower return.
(b) Reward for different numbers of hotspots.

PETS sample-efficnecy tests in simulation



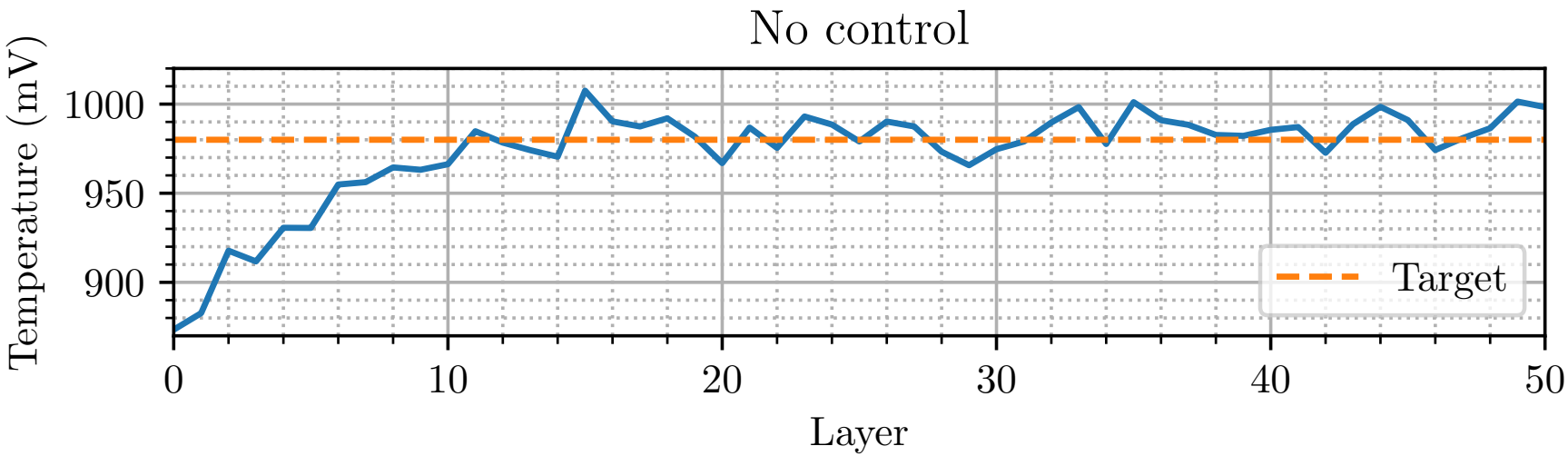
PETS control performance with respect to the number of cubes printed. The solid lines represent the mean return obtained for the 10 runs performed, while the shaded areas represent one standard deviation. "No control" refers to the use of constant optimal parameters.

Controller performance comparison in simulation

Metrics for the Haynes 282 experiment (over 20 trials).

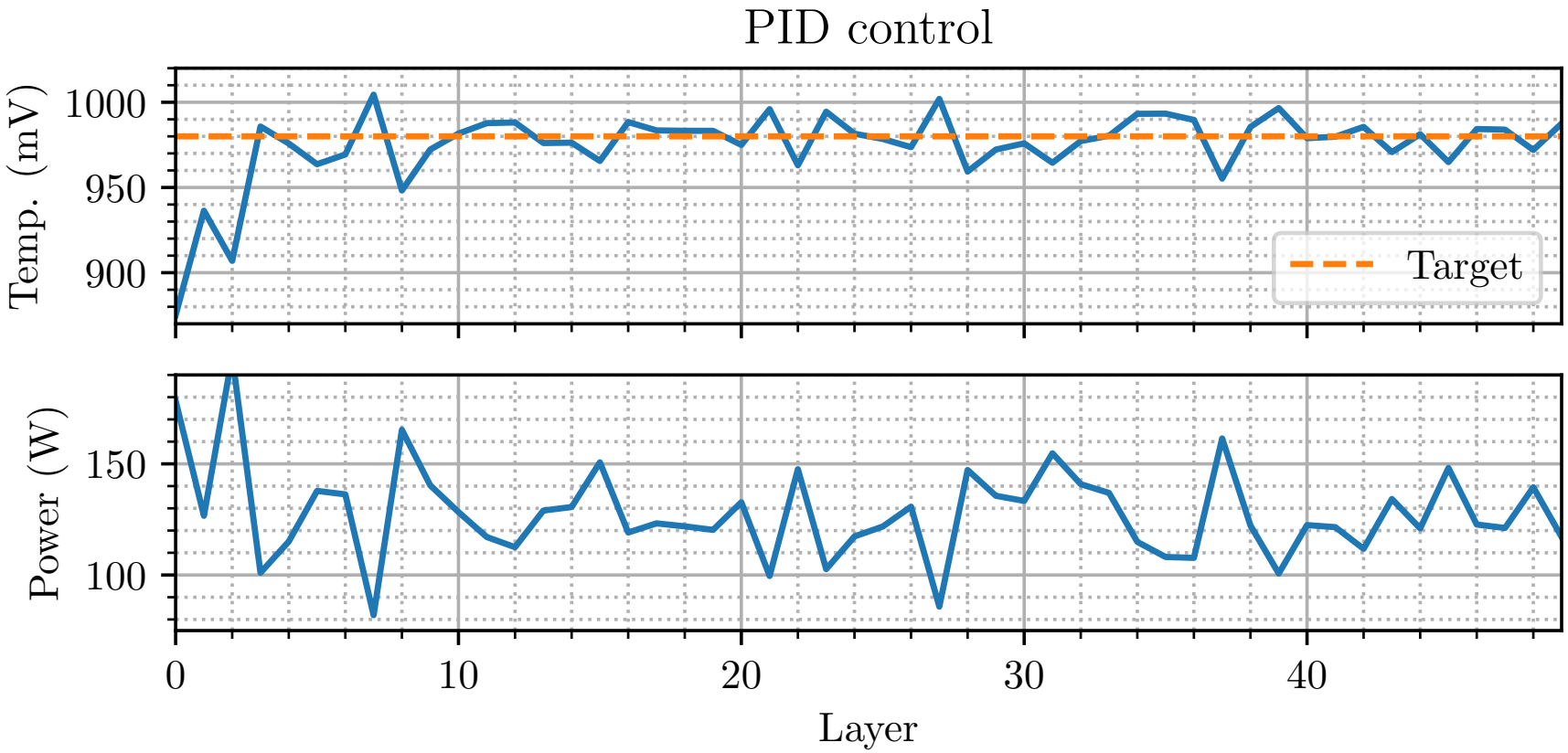
	RMSE	$> +50$ (%)	StdDev	Return
No control	35.1±3.6	1.0±0.4	12.6±0.4	1.21±0.56
PI control	26.6±1.1	1.0±0.6	14.2±0.5	1.55±0.27
PETS control	**24.9±0.6**	**0.1±0.1**	**12.0±0.3**	**4.71±0.92**

Performance of no control in simulation



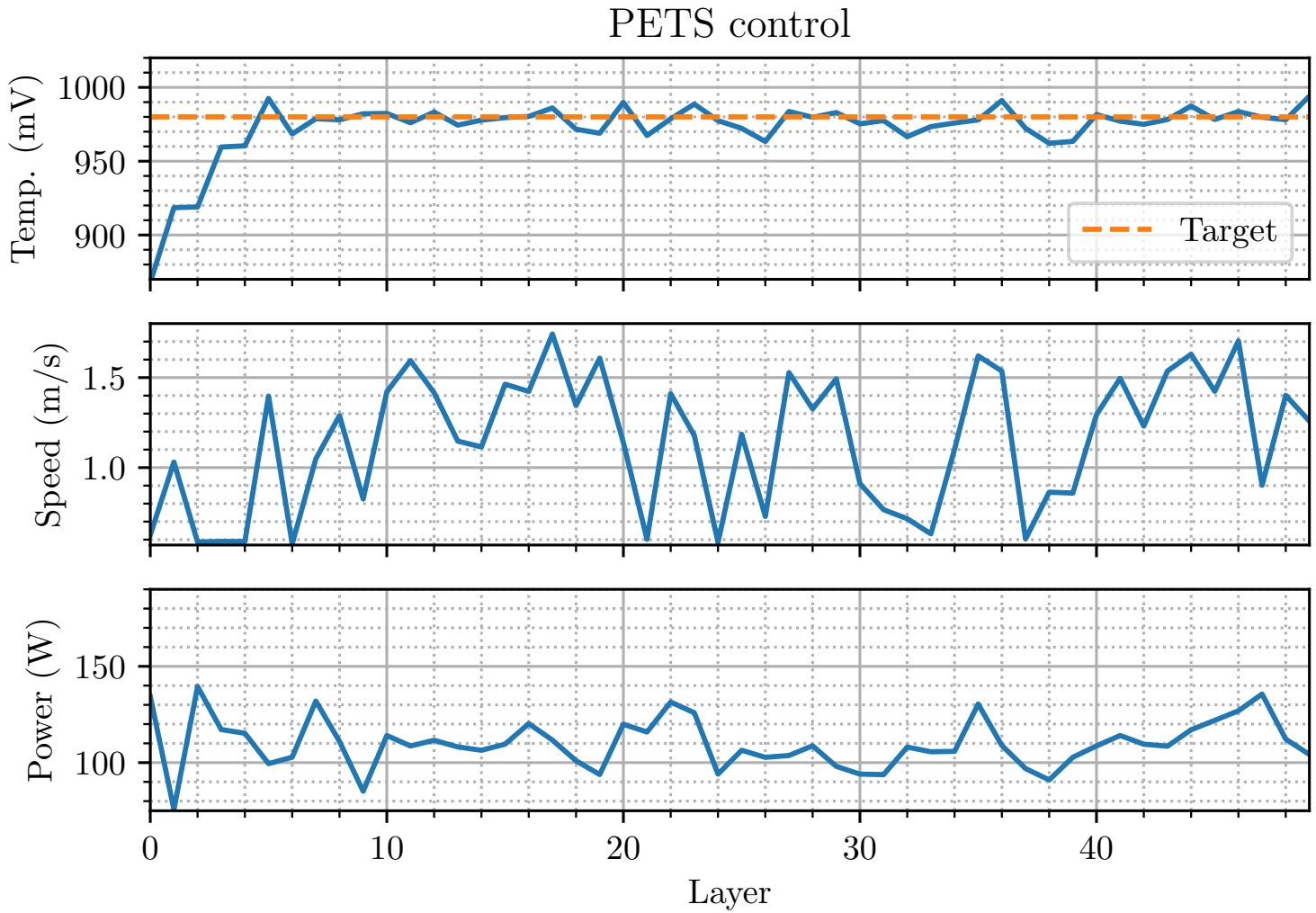
Time response of fixed optimal process parameters.

Performance PID simulation



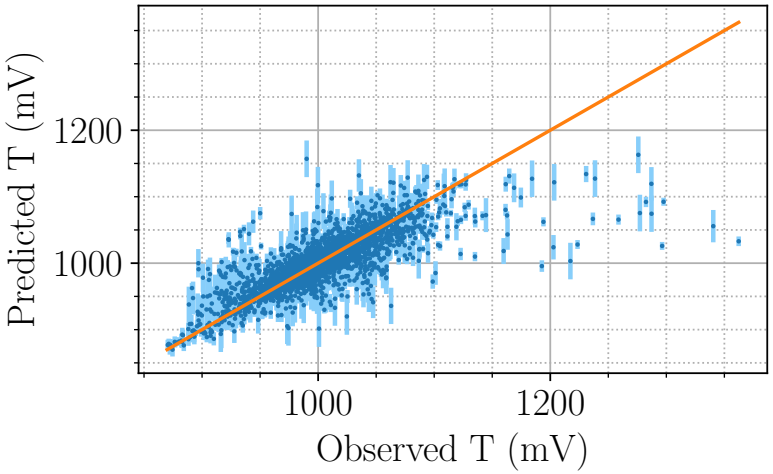
Time response for PID control.

Performance PETS simulation

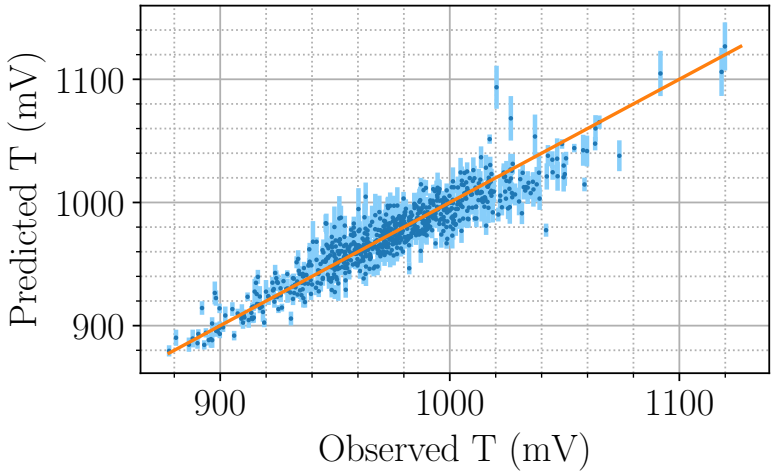


Time response for PETS control.

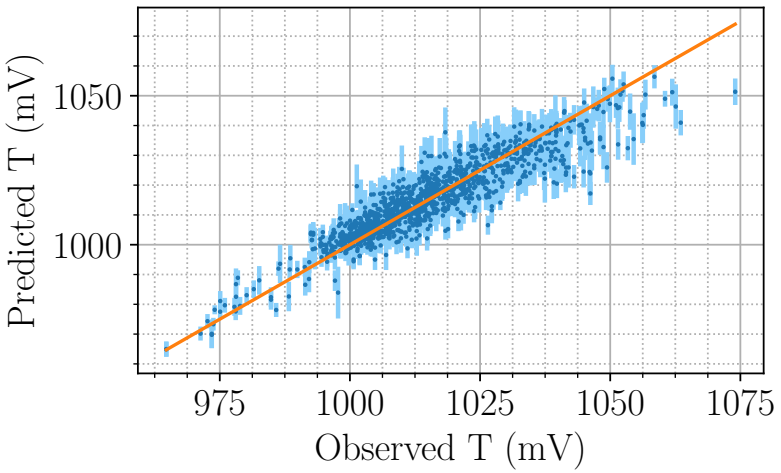
Modelling Plots

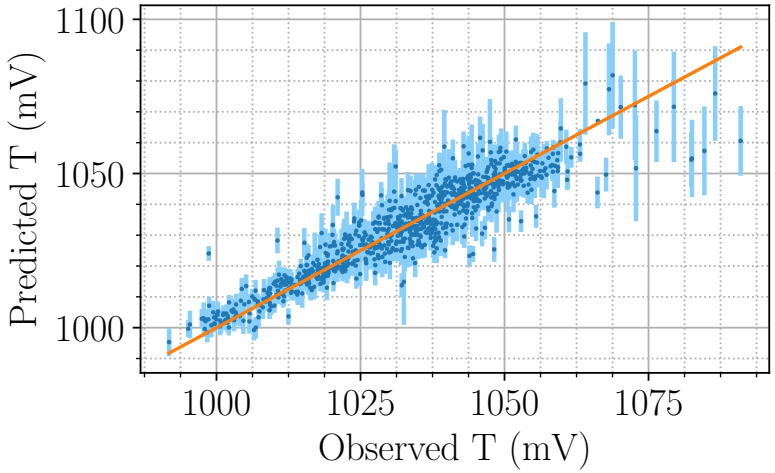


Build 1.1



Build 2.1





Modelling Table

Model performance for the different builds performed.

Build	Datapoints	MSE	R^2
1.1	13972	844±94	0.627±0.021
1.2	4776	621±143	0.675±0.029
2.1	3948	204±38	0.834±0.038
2.2	8780	232±57	0.807±0.035
3	6280	50±5	0.849±0.024
4	4820	54±5	0.840±0.028