

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Optimization of Rastered Beam for Increased Isotope Production at LANSCE

Permalink

<https://escholarship.org/uc/item/66k3p1xv>

Author

Ha, Nicholas

Publication Date

2018

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Optimization of Rastered Beam for Increased Isotope Production at LANSCE

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Electrical Engineering (Intelligent Systems, Robotics, and Control)

by

Nicholas Ha

Committee in charge:

Professor Miroslav Krstic, Chair
Professor Michael Yip, Co-Chair
Professor Nikolay Atanasov

2018

Copyright
Nicholas Ha, 2018
All rights reserved.

The thesis of Nicholas Ha is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Co-Chair

Chair

University of California San Diego

2018

TABLE OF CONTENTS

Signature Page	iii
Table of Contents	iv
List of Figures	vi
List of Tables	ix
Acknowledgements	x
Abstract of the Thesis	xi
Chapter 1	Background	1
	1.1 Introduction to LANSCE and IPF	1
	1.2 Background on Isotope Production	3
	1.3 Isotope Production at IPF	8
	1.4 System Overview of IPF Beamline	12
	1.4.1 Beam pulse structure	13
	1.4.2 IPF's Target and Target Window	13
	1.4.3 Sensing	15
	1.4.4 Control	15
	1.4.5 Protection	16
	1.5 System Summary	16
	1.6 Optimization Method Currently Used	17
Chapter 2	Problem Scope and Formulation	26
Chapter 3	Models	29
	3.1 Beam distribution $I_{\text{ave}}(r; \theta)$	29
	3.2 Temperature $T(r, t; \theta)$	30
	3.3 Simplifications	32
Chapter 4	Methods Overview	34
	4.1 Properties of the Optimization Problem	34
	4.2 Recasting the Optimization Problem, Designing a new Cost	35
	4.3 Solving for Cost Minimization $J(\theta)$	37
	4.3.1 Extremum Seeking	37
	4.3.2 Gradient Descent	43
	4.3.3 Summary: ES versus Gradient Descent	45
	4.4 Alternate Approaches	46
	4.4.1 Genetic Algorithms	46
	4.4.2 Casting as an MDP, solving with Reinforcement Learning	47

	4.5	An Easier Sub-Problem	50
	4.5.1	Gradient Descent	51
	4.5.2	Expectation Maximization	51
	4.6	Summary	52
Chapter 5		Methods Used	54
	5.1	Implementation	54
	5.2	Procedure	55
Chapter 6		Results	60
	6.1	Comparison of Methods	60
	6.2	Results for Bounded ES and Gradient Descent	61
	6.2.1	Nominal system parameters, for different goals	61
	6.2.2	Off-nominal Constraint Values	63
Chapter 7		Conclusions and Future Work	69
Bibliography		71

LIST OF FIGURES

Figure 1.1:	Diagram of LANSCE accelerator. IPF receives diverted H+ 100 MeV beam, while the H- beam continues to be accelerated to 800 MeV before being directed to other experimental areas	2
Figure 1.2:	Proton Emission Tomography (PET) scan of the heart muscle which uses Strontium-82	2
Figure 1.3:	Diagram (not to scale) of main components of the end of the IPF beam-line which uses rastering. Raster magnets (not shown) are used to deflect the beam and draw a raster pattern over the target. The deflection angle is exaggerated here (for IPF, deviation angle from the z axis is	5
Figure 1.4:	Ideal beam distribution for isotope production: uniformly distributed without beam spill on the collimator	6
Figure 1.5:	Beam Profile, with IPF Target and collimator outlined in black and red. In this figure and all other figures, the z-axis is normalized according to the time structure of the beam.	7
Figure 1.6:	Target window temperature and beam distribution simulations at iThemba labs. (beam diameter, raster diameter) are varied as indicated. The dashed line indicates the best combination of the ones shown.	9
Figure 1.7:	Graph showing the proposed rule applied for a spiral raster pattern. The time axis is time that should be spent a radial position, rather than the elapsed time during a pulse of beam.	10
Figure 1.8:	Current Density for original single-circle pattern	11
Figure 1.9:	Current Density for an employed spiral pattern over several pulses, starting at different angles to ensure uniform coverage azimuthally. This is far more uniform than the single circle pattern.	12
Figure 1.10:	Beam Distribution from baseline raster sequence currently being used. During each macropulse, 3.1 circles are drawn, so it's slightly non-radially symmetric.	19
Figure 1.11:	IPF beam time structure	20
Figure 1.12:	Raster sequence used for 2" target	20
Figure 1.13:	Beam Distribution from one macropulse rastered with radius 10.4 mm. Note the overlap; the beam is not perfectly radially symmetric because 3.1 circles are drawn per macropulse. The overlap occurs at different radial locations so the average beam distribution (figure 1.10) is more radially symmetric.	21
Figure 1.14:	the IPF beamline includes steering and raster magnets (for both X and Y directions) and several BPPMs for measuring beam (x,y) position	22
Figure 1.15:	BPPMs measure beam (x,y) at 1 MHz at various z locations along the beamline. The right figure shows the time axis as well.	22
Figure 1.16:	Thermal Simulation heat map for a 3 circle pattern	23

Figure 1.17:	Temperature vs time for 1.2 sec, sampled at the 3 raster radii for a 3 circle pattern. Temperatures at Node A, B and C are plotted in the graph. Note that point A (closest to middle, in blue) has the hottest temperature extremes. This is because smaller raster circles deliver the same amount of beam over a smaller area, increasing the beam density.	23
Figure 1.18:	Block diagram showing components that control beam (x,y) position on target. The components with transients are blue.	24
Figure 1.19:	Magnetic field of raster magnets have transients when changing amplitudes. Here, the raster radius is changed around 0.017 and 0.025 sec, before which, previous transients have long decayed. This makes control of the raster system straightforward.	24
Figure 1.20:	A slice of charge density delivered from a single macropulse when rastered with different radii (i.e. a slice using $y = 0$ of a graph similar to figure 1.13). The beam is Gaussian, so the charge density looks Gaussian far from the center at $r = 0$. For smaller r , it doesn't look Gaussian anymore	25
Figure 4.1:	(copied from [12]) Block diagram for a simple single input cost map $f(\cdot)$. $\hat{\theta}$ is the current estimate of θ^* , and it is perturbed by signal $a \sin(\omega t)$.	39
Figure 4.2:	(copied from [12]) Block diagram for a multi-input cost map $Q(\cdot)$. \hat{G} is the current gradient estimate, $\hat{\theta}$ is the current estimate of θ^* , and it is perturbed by signal $S(t)$.	39
Figure 4.3:	(copied from [15]) Block diagram for Bounded ES scheme for cost $J(\cdot)$. This shows the updates for a single element i of the parameter vector. The cost appearing in the cosine term ensures bounded update rates.	41
Figure 4.4:	(copied from [15]) Illustration of Bounded ES for a 2D parameter vector in with quadratic cost. While the magnitude of update is constant, the turning rate is higher when moving in the wrong direction, and slower when in the correct direction.	42
Figure 5.1:	Raster sequence determined from one trial of Gradient Descent. For comparison, it also shows the baseline and initial guess used for this run. <i>Best design cost</i> in is shown in red but is covered by <i>best true cost</i> . Notice that the sequence hasn't changed much other than the	56
Figure 5.2:	$I_{\text{ave}}(r)$ at the end of the trial. The red dashed line shows the result that had the lowest design cost, an invalid solution because of a constraint violation. The black line shows the best valid solution that was ever encountered during the optimization. Recall that I_{target} , the amount	57
Figure 5.3:	The hottest temperature on the target as a function of time. The points plotted are actually only measured at the end (hottest part) of each macropulse. We see that despite having much more current (as shown in figure 5.2), our solution (in black) does not violate the	57

Figure 5.4:	The cost trajectories as a function of iteration number during the algorithm. Total design cost (J) is plotting along with the component due to current, I_{limit} constraint, T_{limit} constraint, and true cost. The corresponding components for the baseline parameters are	58
Figure 5.5:	Parameter trajectories as a function of iteration number. I_{peak} and σ have markers, while the other curves are raster radii in the sequence. Here we see that most parameters changed smoothly, but I_{peak} stopped abruptly when it ran into a hard boundary that was defined for it. . .	59
Figure 6.1:	Solution obtained from Bounded ES for maximum current goal. The first graph shows the raster sequence. Notice that this shows more deviation from the baseline than for the Gradient Decent example in figure 5.1 . This was always the case. The second graph shows the . .	62
Figure 6.2:	Solution obtained from Gradient Descent for maximum current goal. The circles are slightly larger so that I_{peak} can be increased without overheating the center. There is higher current density at larger radial position r which greatly increases total beam delivery.	63
Figure 6.3:	Solution obtained from Bounded ES for mixed goal. Coverage at the center of the target wasn't compromised in achieving more overall beam delivery.	63
Figure 6.4:	Solution obtained from Gradient Descent for mixed goal. As in figure 6.3 , $I_{\text{ave}}(0)$ wasn't reduced when achieving higher I_{target}	64
Figure 6.5:	Solution obtained from Gradient Descent on a system with simpler constraint on for I_{spill} . This cost function was easier to solve for this particular goal. As expected, this trades off a great deal of current delivery for uniformity. In practice, total beam delivery is more	64
Figure 6.6:	Solution obtained from Bounded ES for mixed goal for system with increased I_{limit}	65
Figure 6.7:	Solution obtained from Gradient Descent for mixed goal for system with increased I_{limit}	66
Figure 6.8:	Solution obtained from Bounded ES for mixed goal for system with increased T_{limit}	66
Figure 6.9:	Solution obtained from Gradient Descent for mixed goal for system with increased T_{limit}	67
Figure 6.10:	Solution obtained from Bounded ES for mixed goal for system with both limits increased.	67
Figure 6.11:	Solution obtained from Gradient Descent for mixed goal for system with both limits increased.	68

LIST OF TABLES

Table 1.1:	Table of key parameters. See chapter 3, equation 3.5 for description of temperature models and parameters. *Raster sequence not listed here, see figure 1.12	17
Table 5.1:	Table of outputs to determine how successful this run was. The final result is the last row. We see that this solution produced had higher I_{peak} and put more current on the target without violating the constraints (1.9 uA and 100 C). T_{max} is at the constraint, and is the limiter	58
Table 6.1:	Summary of results for ES and Gradient Descent (GD) for nominal constraints. All solutions increased I_{peak} in order to achieve higher I_{target} without violating the constraints. The measure of uniformity shown is $\frac{I_{\text{ave}}(0)}{\max_r I_{\text{ave}}(r)}$. For this metric, the baseline couldn't be beat.	62
Table 6.2:	Summary of results for ES and Gradient Descent (GD) for modified constraints. The three cases are higher I_{limit} , then T_{limit} , and then higher both. For baseline, I_{peak} is scaled up while σ and raster sequence are held constant, until one of the constraints is violated.	65

ACKNOWLEDGEMENTS

I'd like to thank the Los Alamos Neutron Science Center's (LANSCE's) Accelerator Operations and Technology-Instrumentation and Controls (AOT-IC) group, as well as the Isotope Production Facility (IPF) team for supporting this work. In particular, thanks to Martin Pieck for his consistent guidance for all of my work conducted on that team, and thanks to Heath Watkins for supporting my other efforts (not discussed here) to upgrade functionality of some of the IPF beamline equipment. Also, thanks to Alex Scheinker from the AOT-Radio Frequency Engineering (RFE) group for his technical assistance in solving the problem addressed in this thesis.

Thanks to my advisor Professor Miroslav Krstic for his guidance in solving the problem addressed in this thesis, and in ensuring it meets standards of quality academic research.

ABSTRACT OF THE THESIS

Optimization of Rastered Beam for Increased Isotope Production at LANSCE

by

Nicholas Ha

Master of Science in Electrical Engineering (Intelligent Systems, Robotics, and Control)

University of California San Diego, 2018

Professor Miroslav Krstic, Chair
Professor Michael Yip, Co-Chair

The Isotope Production Facility (IPF) receives pulsed H⁺ beam from the accelerator at the Los Alamos Neutron Science Center (LANSCE) to produce isotopes for various research and medical uses. Work over 2 summers has been conducted to improve diagnostics, reliability, and control for equipment on the IPF beamline, as well as to optimize beam delivered to IPF to increase isotope yield. This second goal is the subject of this document. Specifically, it is desired to maximize beam delivery onto a target without risking damage to the target window or spill too much beam outside a specified aperture. A specified beam profile and repeating beam raster pattern of concentric circles is currently being

used to achieve this. The goal addressed here is to determine an improved beam profile and raster pattern. While a specific answer is needed for certain operation conditions, a general method is even better. A constrained optimization problem is formulated, and several approaches for solving it are discussed. It is found that with appropriate design of cost function, a Gradient Descent algorithm is effective at making improvements to a good initial guess. A bounded Extremum Seeking algorithm also shows some effectiveness, while some unbounded versions did not show effectiveness for this problem. The problem formulation, models, methods, results, are presented. They show significant potential to advance IPF's goals of higher isotope production yield.

Chapter 1

Background

In this chapter, we will give an overview of some previous work on beam optimization for isotope production, describe characteristics specific to LANSCE and IPF, and describe the method currently used for beam optimization at LANSCE.

1.1 Introduction to LANSCE and IPF

The Los Alamos Neutron Science Center (LANSCE) is home to a particle accelerator that provides beam to multiple research facilities including the Isotope Production Facility (IPF), Ultra Cold Neutron (UCN), Proton Radiography, Lujan, and Weapons Neutron Research (WNR) Centers. There are in total 5 targets, and multiple beam “flavors” which refer to different characteristics of beam modes. All receive H^- except for IPF, which receives H^+ beam. Each are generated by respective injectors, then they get bunched and accelerated down the beam pipe (which is under vacuum) to different energy levels: typically 100 MeV for IPF and 800 MeV for the other facilities, as shown in **figure 1.1**. IPF is a facility that irradiates various targets such as Rubidium metal or Rubidium Chloride targets to create isotopes like Strontium-82. This is used in medical applications such as cardiac imaging (**figure 1.2**). It is desired to use the beam for both production of

isotopes (e.g. Strontium-82, Germanium-68, Sodium-32), and for research and development purposes for new applications (e.g. Actinium-225). In general, the desire to increase the isotope production yield (i.e. production rate) to maximize the efficiency of beam time.

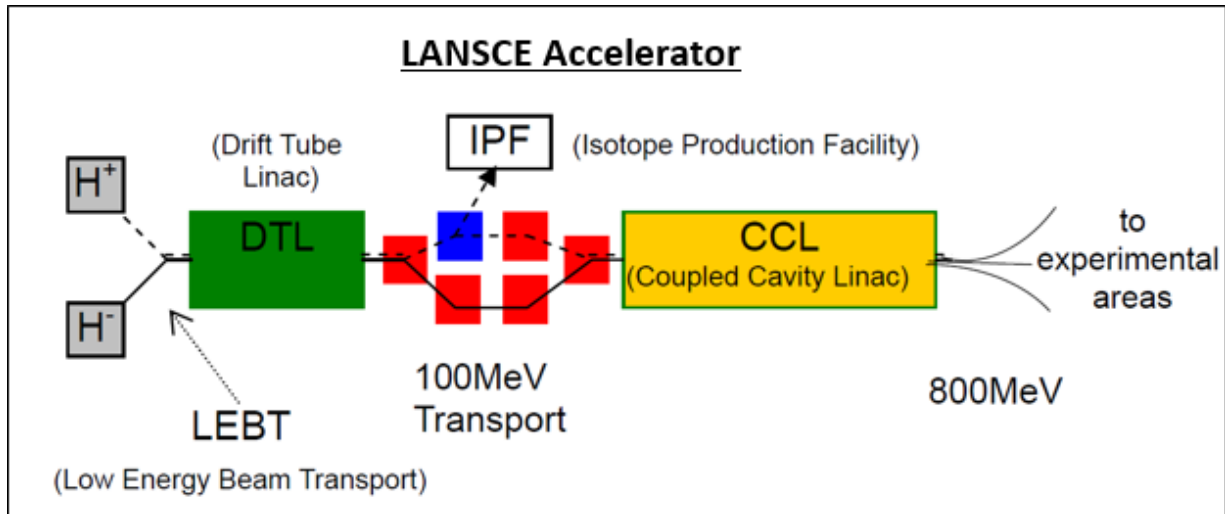


Figure 1.1: Diagram of LANSCE accelerator. IPF receives diverted H^+ 100 MeV beam, while the H^- beam continues to be accelerated to 800 MeV before being directed to other experimental areas

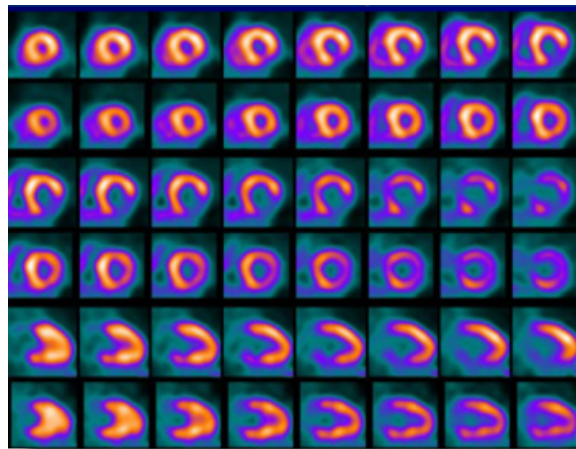


Figure 1.2: Proton Emission Tomography (PET) scan of the heart muscle which uses Strontium-82

1.2 Background on Isotope Production

Bombardment of targets by protons has been a standard method for production of various isotopes for a long time. Depending on specific goals (e.g. quantity vs purity) and materials being used, methods vary greatly. However, many setups have a structure similar to the diagram in **figure 1.3**. The source of protons is a pulsed beam from a proton accelerator, and a rigid or foil “window” of some material (here, Inconel) separates the accelerator pipe (under vacuum) from the target (which is typically submerged in water for cooling). Often, there is a ring-shaped *collimator* upstream which absorbs excess beam that doesn’t fall on the target. This window is subject to thermo-mechanical stress since it’s the barrier between the vacuum of the beam-pipe and water in vessel holding the target. Since the beam passes through the window, the window will also absorb some of the energy and heat up. In addition, precautions must be taken to make sure the powerful beam is stopped by the targets, collimator, or beam-stop. This means the characteristics of the beam and the designs of the target window, beam stop, and collimator must be carefully chosen to deliver beam without damaging the equipment involved. Also, adequate control and protection systems are needed to keep the beam on target, surrounding equipment protected, and personnel safe. This document is concerned with maximizing isotope production by optimizing parameters of a beam whose fundamental characteristics have been determined in conjunction with the rest of the system design. This problem can be summarized in the following form:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \quad P(\theta) \quad (1.1)$$

$$\text{such that} \quad I_{\text{spill}} < I_{\text{limit}} \quad (1.2)$$

$$T(x, y, t; \theta) < T_{\text{limit}} \quad \forall x, y, t \quad (1.3)$$

- θ consists of parameters that describe the beam.
- $P(\theta)$ is the isotope production rate, as function of theta.
- I_{spill} is the average total current falling on the collimator.
- I_{limit} is the maximum current that the collimator can take in steady state. It is a function of the design of the collimator and its cooling system.
- $T(x, y, t; \theta)$ is the temperature of the beam window at coordinate (x, y) at time t and parametrized by θ

Solving the above problem is the ultimate goal, but function $P(\theta)$ represents a great deal of un-modeled or poorly modeled behavior. Standard model-based optimization methods would not work. Generally speaking, for all other factors (target material, chemical handling process, etc) equal, more current on the target results in more isotopes produced. In addition, uniformity of coverage is helpful for isotope production. An example uniform beam distribution is shown in (**figure 1.4**). These qualities allows maximal interaction with and utilization of all of the target material. However, the exact importance of quantity vs uniformity isn't well-understood or modeled. In order to keep the beam window from overheating, more uniform distributions tend to do better since temperature extremes tend to be less severe.

With this in consideration, the problem formulation can be summarized by

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \quad \alpha I_{\text{target}}(\theta) + \beta f(I_{\text{ave}}(x, y; \theta)) \quad (1.4)$$

$$\text{such that} \quad I_{\text{spill}} < I_{\text{limit}} \quad (1.5)$$

$$T(x, y, t; \theta) < T_{\text{limit}} \quad \forall x, y, t \quad (1.6)$$

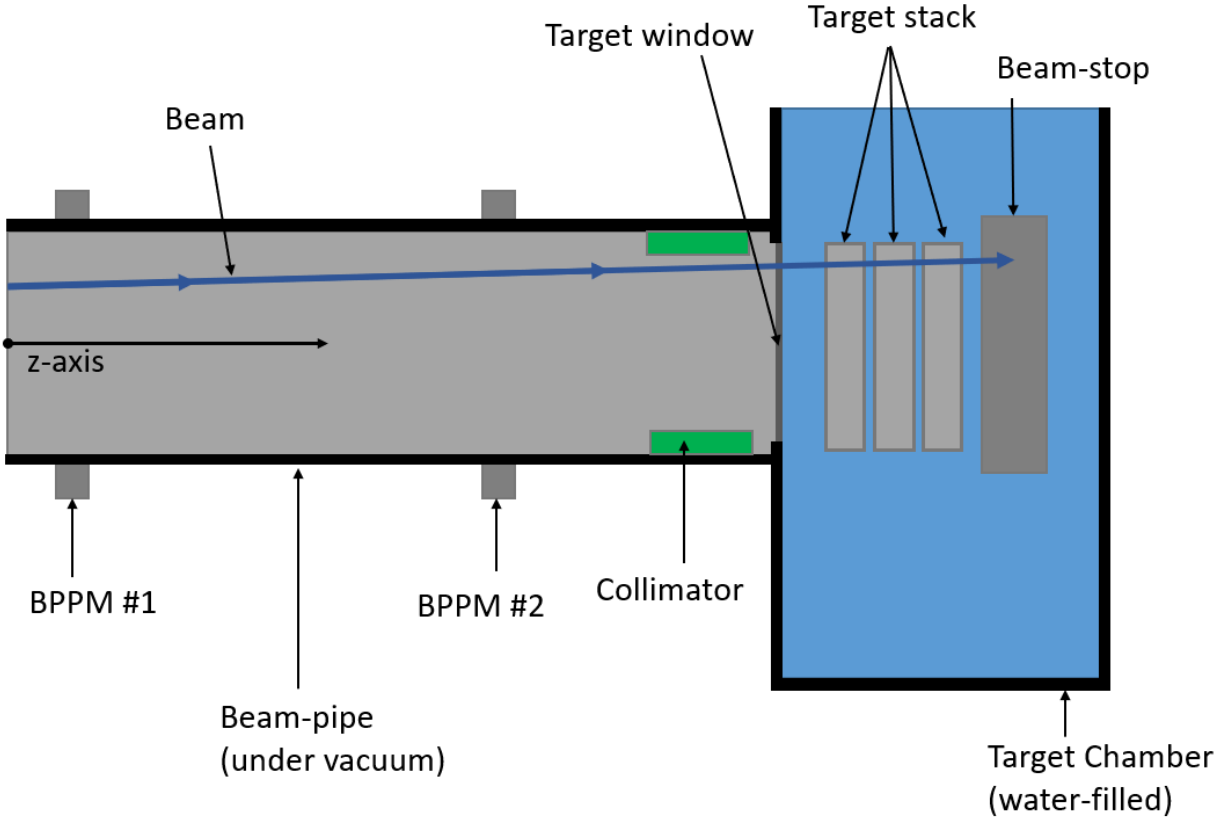


Figure 1.3: Diagram (not to scale) of main components of the end of the IPF beam-line which uses rastering. Raster magnets (not shown) are used to deflect the beam and draw a raster pattern over the target. The deflection angle is exaggerated here (for IPF, deviation angle from the z axis is 1×10^{-4} rad). Beam travels from left to right, losing energy as it passes through the beam window, target stack, and finally ending at the beam-stop. The collimator is a ring that absorbs any excess beam that are off-target. Beam Position and Phase Monitors (BPPMs) are sensors that will be described in section 1.4.3

with additional terms

- $I_{\text{target}}(\theta)$ is the average total current delivered onto the target, parametrized by θ
- $I_{\text{ave}}(x, y; \theta)$ is the average current density delivered at coordinates (x, y) , parametrized by θ
- $f(\cdot)$ is some measure of uniformity
- α, β are coefficients that weigh the relative importance of total current vs uniformity

This is the problem form addressed by most previous researchers for this problem (possibly with additional constraints based on their particular facility).

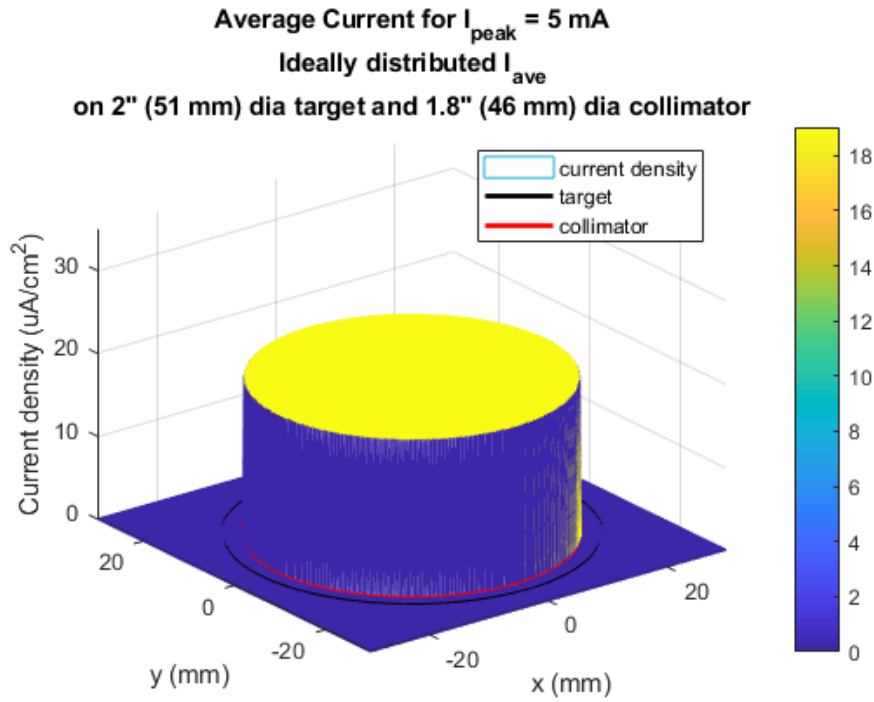


Figure 1.4: Ideal beam distribution for isotope production: uniformly distributed without beam spill on the collimator

Because the proton beam consists of like charges traveling down the beam-pipe, the beam spreads out. Therefore the current density in the plane orthogonal to beam travel is not uniform, but well-approximated by a 2D-Gaussian distribution [1]. A typical beam distribution from the LANSCE accelerator is shown in **figure 1.5**. To achieve beam uniformity on the target, two general methods are employed: beam de-focusing and beam rastering.

De-focusing involves spreading out the beam much wider than the target, so that the middle section that reaches the target is roughly uniform in current density. A collimator is used to absorb the excess beam that doesn't fall onto the target. This method is very wasteful, since so much of the beam ends up on the collimator instead of the target. For

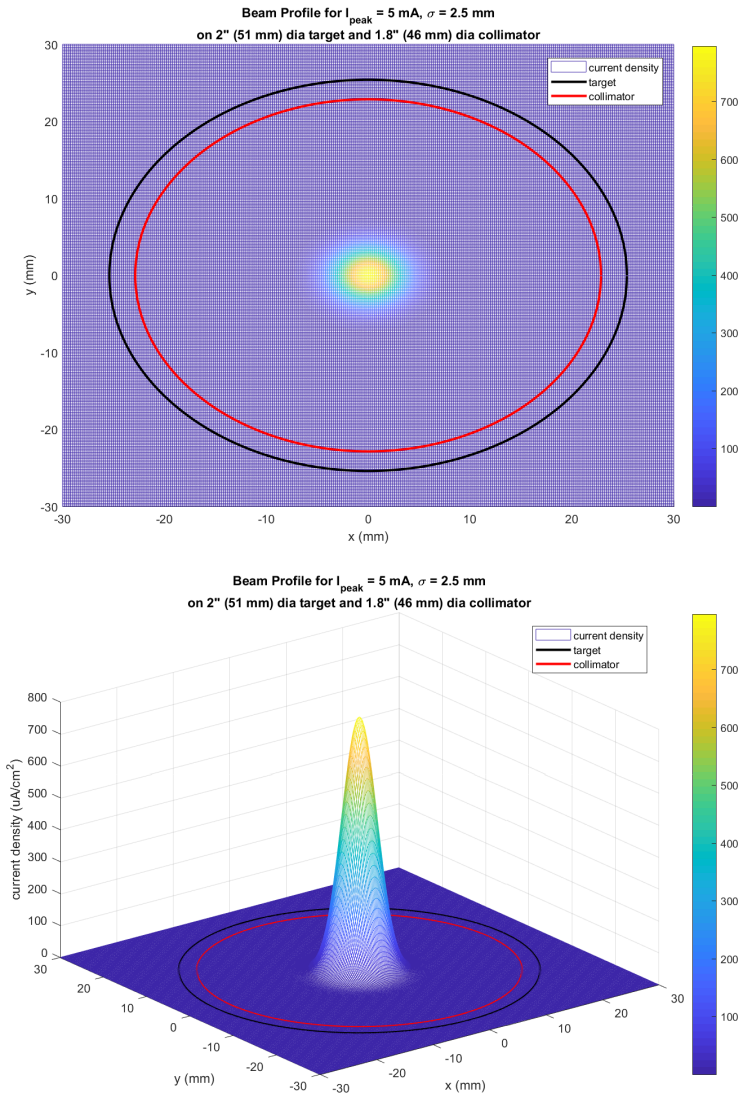


Figure 1.5: Beam Profile, with IPF Target and collimator outlined in black and red. In this figure and all other figures, the z-axis is normalized according to the time structure of the beam.

example, in one application at Harvard, only the middle 5% of the beam is used and 95% of the beam is wasted [1]. Another de-focusing based method uses an aperture along with a carefully shaped beam-stop before the target to block out the middle part of the beam, allowing it to spread back out and fill in the middle section again. With the correct geometry, this method is highly effective at achieving uniform beam distribution on the

target and more efficient than the method above. Still, a great deal of current is absorbed by the beam-stop. For higher current applications, this may be too much beam for any material to absorb without overheating, and it is still not as efficient as it could be. In addition, it is often difficult to maintain wide beams for high currents [2].

Beam rastering involves using a beam that is narrow relative to the target, then *rastering* (drawing) it over the target using a specified pattern. While often more complex, this is much more efficient. Beam rastering for isotope production was proposed as early as 1985, in which it is noted that the raster frequency must be “far greater than the reciprocal of the thermal rise time (of the target window)” [3]. iThemba Labs in South Africa is an example of a facility similar to LANSCE’s IPF that uses beam rastering. [2]. With the goal of maximum current subject to thermal constraints, the solution employed there uses a single circle raster pattern. Given a fixed raster frequency, the remaining 2 parameters to determine are beam width and raster radius. The method for finding the optimal combination of these parameters was trial and error by simulating the current density and temperature. The resulting optimum found was with a raster diameter about 30% larger than the beam diameter. According to [2] this result is similar to results found in TRIUMF, another Isotope Production Facility in Canada. **Figure 1.6** shows graphs from [2] for current density and simulated temperature for different combinations of raster and beam diameters.

1.3 Isotope Production at IPF

In the past, LANSCE’s IPF had used beam rastering with a single raster radius, similarly to iThemba Labs. However, LANSCE recently experimented with more complex patterns. First, it was found by geometric analysis that in order to achieve uniform current density, there should be roughly a square root dependence between time spent at a radius

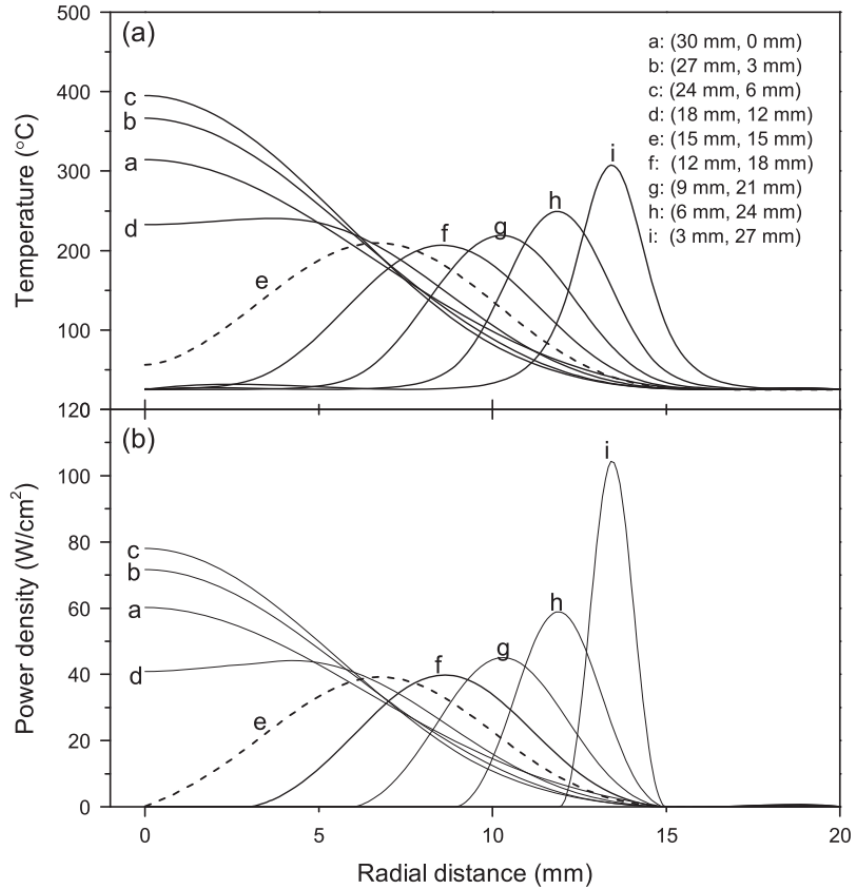


Figure 1.6: Target window temperature and beam distribution simulations at iThemba labs. (beam diameter, raster diameter) are varied as indicated. The dashed line indicates the best combination of the ones shown.

and the radius (i.e. $R(t) \propto \sqrt{\tau}$, τ = a “unit” amount of time [4]. This is illustrated in **figure 1.7** and is similar to a result found in [5]. Using this rule, they determined some candidate patterns consisting of spirals drawn at a constant frequency. Unfortunately this rule does not account for beam width, so this insightful but simple rule fails at smaller radii where beam width plays a bigger factor. More complicated optimization through trial and error was needed to complete the patterns near the center of the target, and it was left un-optimized. The temperature constraint was checked later, rather than during the optimization. Results for the spiral are still very good (**figure 1.9**), especially compared to the single circle pattern (**figure 1.8**).

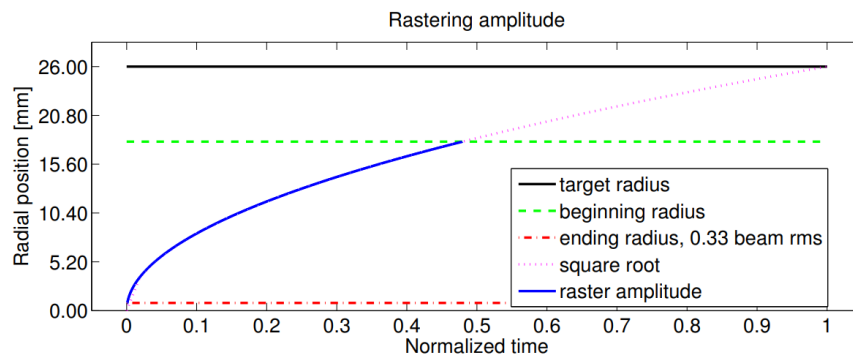


Figure 1.7: Graph showing the proposed rule applied for a spiral raster pattern. The time axis is time that should be spent a radial position, rather than the elapsed time during a pulse of beam.

Currently, LANSCE’s IPF uses a raster sequences with radii selected from 3 possible radii for 2 inch diameter targets and 5 possible radii for 2.5 inch diameter targets. The resulting beam distribution for 2 inch targets is shown in **figure 1.10** These patterns are chosen over patterns used in [4] and [6] for two reasons. First, they cover the edges of the target better(**figure 1.10 vs figure 1.9**). Secondly, it was more difficult to define and detect a dangerous fault condition on the raster system for spirals than for a raster system with concentric circles.

At IPF, targets are irradiated in *run cycles* that last several days or weeks. After a run cycle, the targets are carefully removed for chemical processing at on-site hot-cell processing facilities. The entire process from beginning of the run-cycle to extraction of isotopes is on the timescale of days to weeks. Since the process of trying a new beam pattern and evaluating its efficiency takes a very long time and is subject to many unmeasured variables, beam patterns are rarely changed after being set. Improvements to beam patterns are typically done by proposing a desired beam distribution and trying to find the parameters (peak current, beam width, and raster sequence) that achieve the desired distribution.

At this point it may be useful to address potentially confusing terminology. *Current*

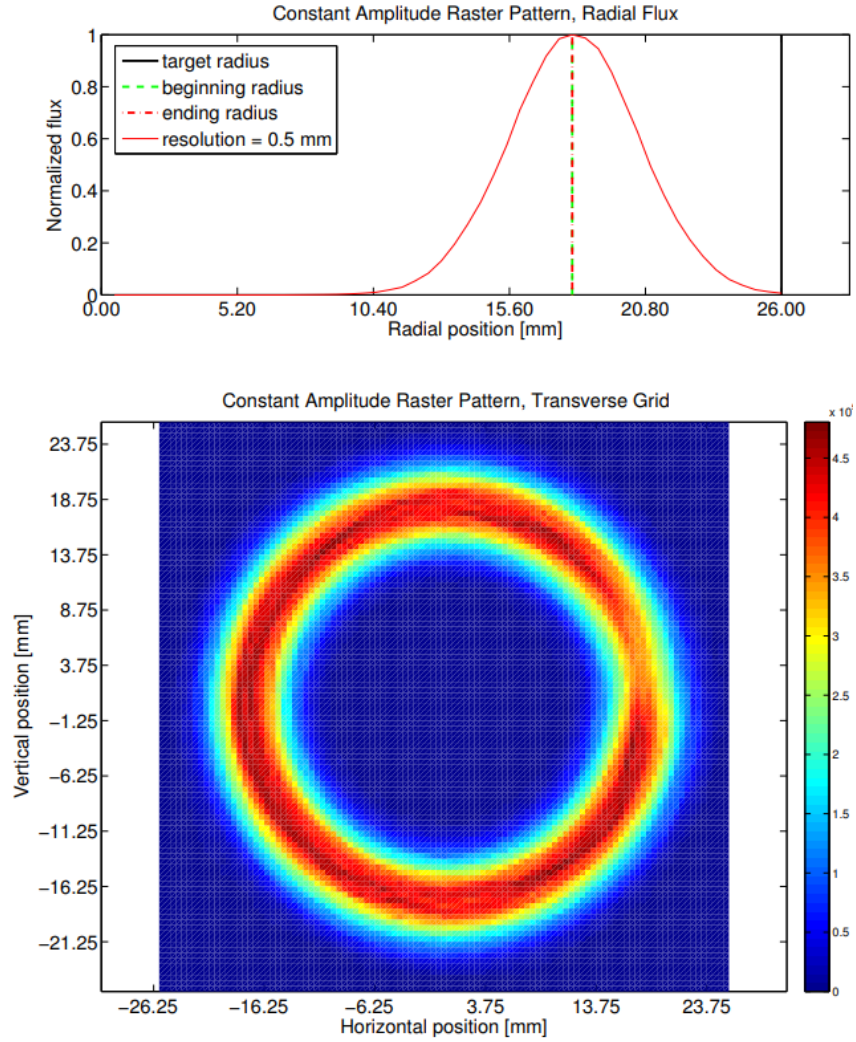


Figure 1.8: Current Density for original single-circle pattern

in this context refers to the total number of charges (i.e. protons), while energy refers to the energy of each individual charge. Since each particle is a proton with fixed mass, energy and speed are directly related as: $v = c \times \sqrt{1 - \frac{1}{(1 + \frac{E}{m})^2}}$ where c is the speed of light, E is beam energy, and m is the rest mass of a proton (938 MeV). Beam and current will be used interchangeably but mean the same thing (i.e. more beam = more current, rather than higher energy). Energy is fixed at 100 MeV for the purposes of our study, though 41 and 72 MeV are used in special cases. As a convention, the beam travels along the z axis, and x, y axes span the face of the circular target.

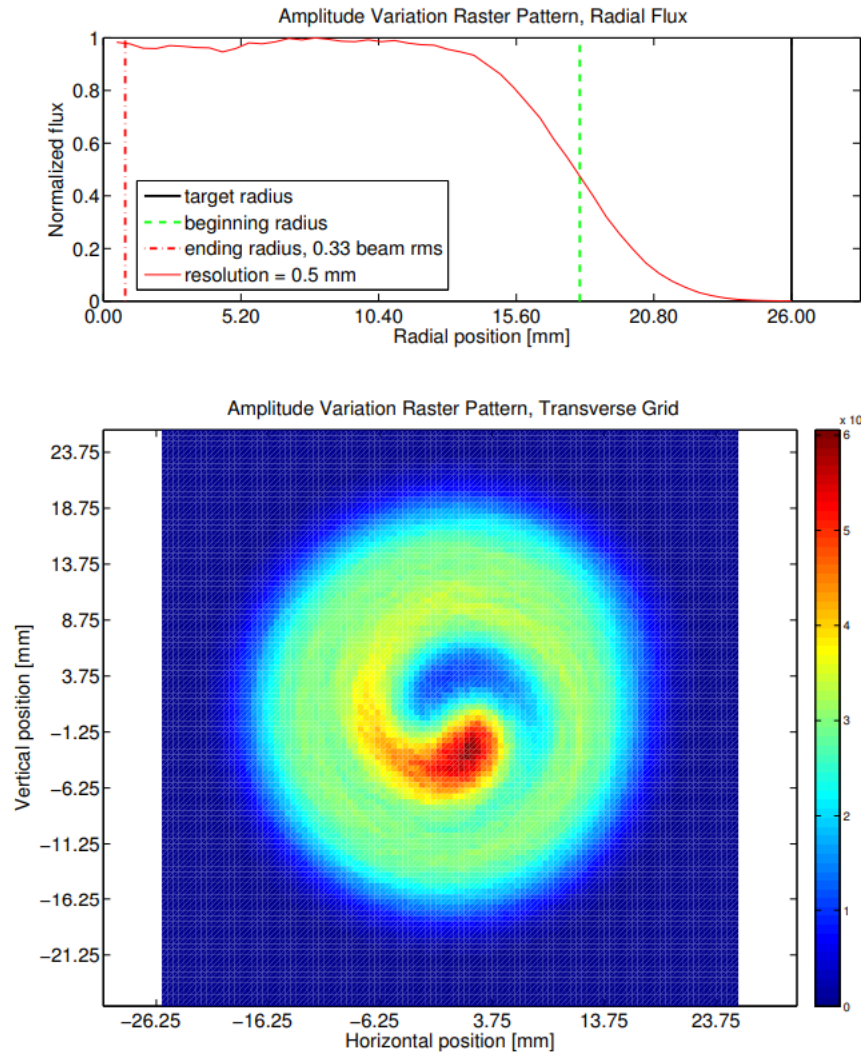


Figure 1.9: Current Density for an employed spiral pattern over several pulses, starting at different angles to ensure uniform coverage azimuthally. This is far more uniform than the single circle pattern.

1.4 System Overview of IPF Beamline

This section will detail specific characteristics of the IPF Beamline.

1.4.1 Beam pulse structure

Time Structure

IPF receives *macropulses* of 100 MeV protons at 120 Hz which is based on zero crossings of standard 60 Hz AC power from the facility. However, every 6th pulse is missing due to the operation of the accelerator, so it can also be considered a non-equally-spaced 100 Hz square wave. Each macropulse is approximately 625 μs long and is composed of 5 ns micropulses. These micropulses have very small magnitude variation so their effect is ignored. For the purposes of this study, we assume the beam structure shown in the **figure 1.11** (which is fairly accurate when beam is running properly). The vertical axis is the total current in the pulse (volume under the surface in **figure 1.5**). It is assumed that during correct operation, there is no variation in energy or current during a macropulse. Also, the rising and falling edges are considered instantaneous.

Spatial Structure (*Beam Profile*)

As mentioned in the previous section, the beam profile is best modeled as a 2D Gaussian, as shown in **figure 1.5**. Current is the volume under the surface, while the vertical axis values are current density in $\mu\text{A}/\text{cm}^2$. Typically the beam peak current $I_{\text{peak}} \approx 5$ mA and the beam widths $\sigma_x \approx \sigma_y = \sigma = 2.5$ mm. *Beam profile* will refer to current density delivered from a single unrastered macropulse (**figure 1.5**). *Beam distribution* will refer to I_{ave} , the current averaged over a long time window after the beam is rastered following a specified raster sequence (**figure 1.10**).

1.4.2 IPF's Target and Target Window

IPF's target area resembles the setup described earlier and illustrated in **figure 1.3**. There is stack of disc shaped targets submerged in water for cooling purposes, and they are

either 2 or 2.5 inches (51 or 64 mm) in diameter, much wider than the beam. The beam comes from the accelerator, passes through the beam window, and passes through each target. In each step, it loses energy through the medium it passes through. Once passes the last target, the protons are significantly slowed down or stopped. A beam-stop located behind the target stack serves as a barrier that absorbs any remaining protons.

There is a metal beam window made of Inconel that separates the water in IPF chamber from the vacuum of the beam pipe. It is extremely crucial that this beam window does not break, or it would spill contaminated water into the beam pipe, resulting in extremely costly and time-consuming repairs and clean-ups. While the pressure and temperature differentials induce great stress, heating is the primary concern. The primary failure mode for the window would be overheating due to water boiling off and resulting in a thermal runaway.

There is a ring-shaped collimator in front of the beam window as well. The primary purpose is to absorb excess beam that won't hit the target, so that it doesn't continue and damage either the beam pipe walls or objects behind the target. This particular collimator is unique because it has an adjustable aperture and is separated into 4 sections, each outfitted with a current monitor. This provides some diagnostic information about how much beam is spilling onto each section, indicating if the beam is not centered or if it's being rastered incorrectly.

The primary failure modes are target window overheating and collimator over-current, so they are addressed here. Other possible failure modes are breakage of the beam window through stress from a pressure differential, and a phenomenon called *ratcheting* where the targets swell and fuse together. These effects are not well modeled or understood so they are ignored for this analysis.

1.4.3 Sensing

In-line, invasive diagnostic equipment called Harps and Wire Scanners are needed to measure the beam profile. They are used when beam is at low current and low frequency (roughly 4 Hz), and cannot be used when beam is in production mode (100 Hz). Therefore they do not provide real-time measurements. However, the beam typically remains stable so that these measurements remain accurate despite the change in frequency. Non-invasive Beam Position and Phase Monitors (BPPMs) cannot detect beam profile, but they measure beam centroid very accurately and quickly at 1 MHz (625 measurements per 625 μ s macropulse).

There is no measurement of temperature on the beam window. Instead, detailed simulation studies have been done previously during the design of the beam window (**figures 1.16, 1.17**).

1.4.4 Control

Beam profile cannot be automatically controlled online once a run-cycle begins. I_{peak} is set by an operator, while beam width σ is set using a long process of manually focusing/de-focusing quadrupole magnets and reading measurements from a Harp.

Beam (x, y) position is controlled using steering and raster magnets. Horizontal and vertical steering magnets keep the beam centered, while raster magnets draw the concentric circles. Raster magnets have power supplies operated at 4.95 kHz, with amplitude controlling radius and with fixed phase shift of approximately 90 degrees. Due to 625 us and 4.95 kHz, just over 3 circles are drawn on the target per macropulse (**figures 1.13**) For simplification, radii are not changed during the macropulse, but right after. This way, any system transients (**figures 1.18, 1.19**) have sufficiently decayed. Control of raster (and steering) is done in open-loop, and fine adjustments are done during the accelerator

tune-up period. This generally works, since this section of the accelerator is a very well controlled environment. However work has been done developing monitoring and real-time feedback control, enabled by the greatly improved equipment which provides faster BPPM data.

The raster sequence cannot currently be changed online, however changes to raster controller software could be made to enable this if desired. It would simply replace a pre-programmed radius sequence in its memory with a sequence from an external source that determines the best raster sequence in real-time.

1.4.5 Protection

In addition to the control system, a fail-safe protection system of interlocks is used to turn off beam extremely quickly (approx. $200 \mu\text{s}$) when errant conditions are detected. In theory, we could use this system to deliberately cut some macropulses short and use that as another way to arrive at an optimal IPF beam. However, this is not what the protection system was designed for and operating this way would be too risky. We could also change radii during a macropulse, resulting in patterns like spirals. However, for this problem we do not address these 2 options and limit ourselves to concentric circles since, as mentioned before, previous studies indicate these may be better [4], [6].

1.5 System Summary

I_{peak} , σ , and raster sequence are the IPF beam parameters that define the IPF beam and determine the average delivered current I_{ave} and temperature T . These 3 parameters make up the vector θ chosen to shape these 2 quantities as desired in order to solve the problem defined in **equations 1.4 - 1.6**, then they are set by operators (beam profile) or programmed into the raster controller (raster sequence). The raster sequence has the

possibility of being modified online, while beam profile cannot be. **Table 1.1** lists key parameters for the IPF beamline.

Table 1.1: Table of key parameters. See chapter 3, **equation 3.5** for description of temperature models and parameters. *Raster sequence not listed here, see **figure 1.12**

Parameter Type	Parameter	Symbol	Value	Unit
Fixed				
	RepRate	F	120	Hz
	Macropulse length	L	625	μs
	Raster frequency	f	4950	Hz
	Ambient temperature	T_{ambient}	22	C
	Beam window specific heat coefficient	α	1.22×10^6	$\frac{\text{C}}{\text{s}\cdot\mu\text{A}/\text{mm}^2}$
	Beam window cooling coefficient	τ	8.37	s^{-1}
Target Dependent				
	Target radius	-	25 or 32	mm
	Collimator radius	r_{limit}	23 or 29	mm
Constraints				
	Max temp	T_{limit}	100	C
	Max current spill on collimator	I_{limit}	0.4 to 2.0	μA
Baseline Beam Parameters				
	Peak current	I_{peak}	5	mA
	Beam width	σ	2.5	mm
	Raster sequence	\mathbf{R}	*	mm

1.6 Optimization Method Currently Used

As stated before, the goal is for I_{ave} to be well-distributed and maximal within specified bounds, and without too much current spilling onto the collimator and without beam window temperature exceeding T_{limit} . The baseline IPF beam optimization routine to pick IPF beam parameters following is educated trial and error using intuition.

Intuitively, higher I_{peak} means more total current on target, but it has higher max current density so it further risks violating constraints. Lower σ allows more current to be distributed near edges without spilling too much onto the collimator, but it increases peak current density so it risks overheating parts of the target. We need small and large radii to cover the target well. Smaller radii deposit much higher current density per pulse than larger circles, since they cover less area during a macropulse, and beam “spills” over more from the opposite side of the circle (**figure 1.20**).

To simplify the problem, σ is fixed. Then, a number of distinct allowable radii is set, and a sequence is derived from only these radii in order to make I_{ave} look like a uniform distribution up to the collimator aperture. For example, for the 2” target, radii are picked from $r_1 = 4.6$, $r_2 = 10.4$, and $r_3 = 16.4$ mm. Out of 21 total pulses, 3 of them are r_1 , 7 are r_2 , and 11 are r_3 .

Then to ensure that temperature does not exceed limits, the order is determined by spacing the smaller radii as far apart as possible, and then I_{peak} is reduced if necessary.

Results for the 2” target are depicted in **figures 1.12, 1.10, and 1.17**. While this method is effective at utilizing our intuition about the problem and does find fairly good solutions, it is not rigorous and structured enough to search the entire parameter space for more optimal solutions.

In this chapter we introduced the problem of beam optimization for isotope production, described the relevant components of the IPF beamline, and illustrated important quantities that describe the goals and constraints of this problem. With this background information in mind, and with the baseline optimization method as a benchmark, we can more formally define the problem and scope at hand.

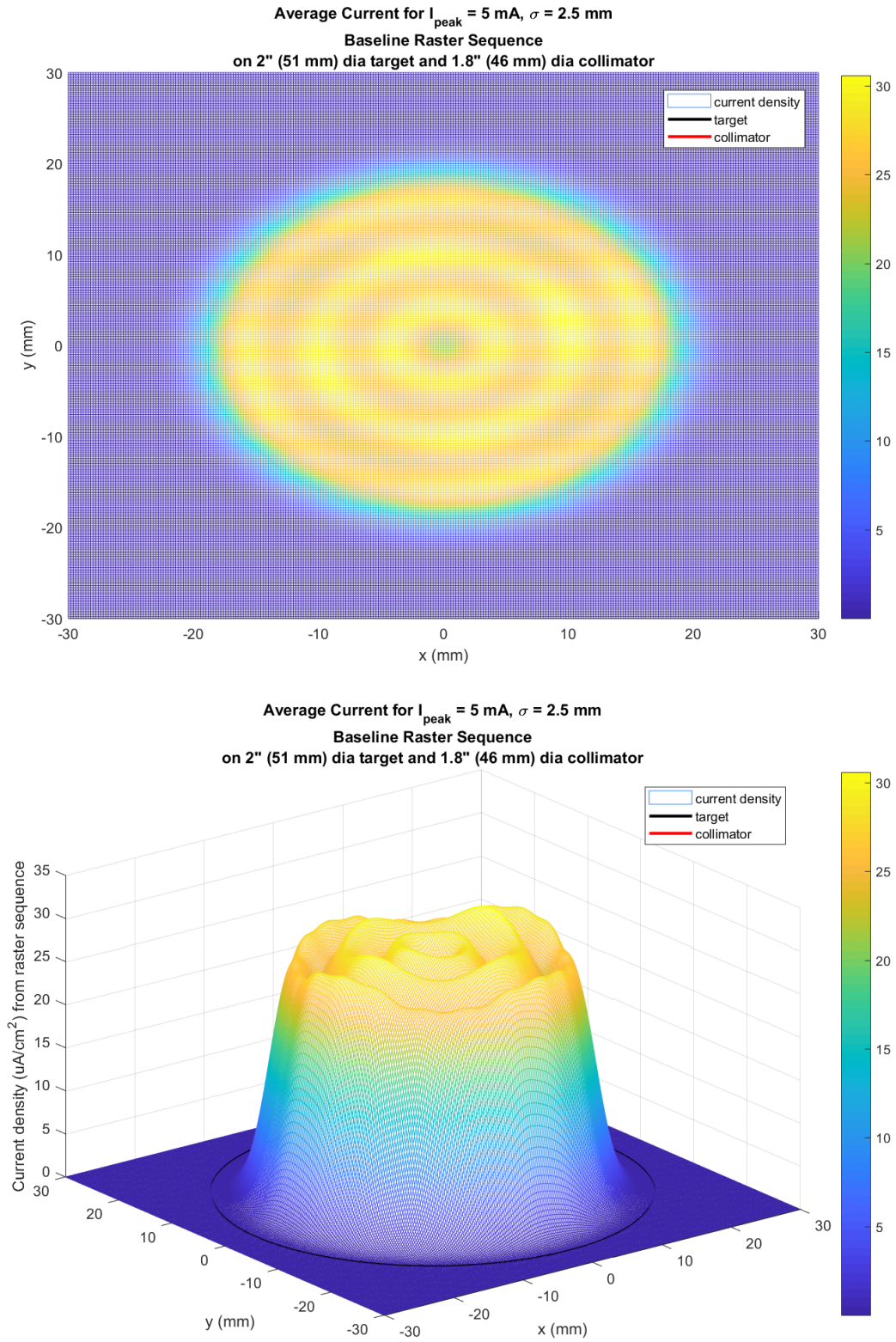


Figure 1.10: Beam Distribution from baseline raster sequence currently being used. During each macropulse, 3.1 circles are drawn, so it's slightly non-radially symmetric.

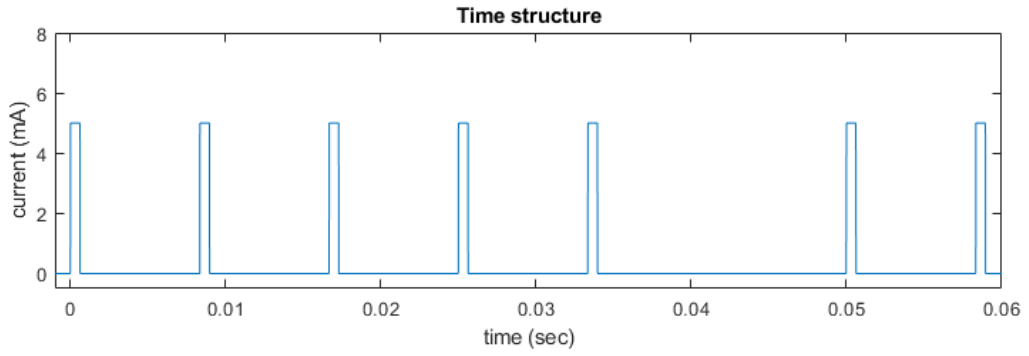


Figure 1.11: IPF beam time structure

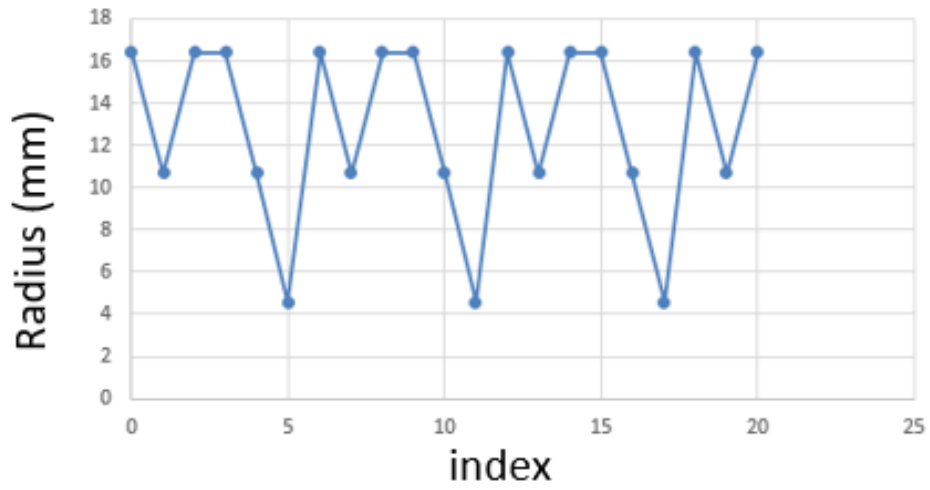


Figure 1.12: Raster sequence used for 2'' target

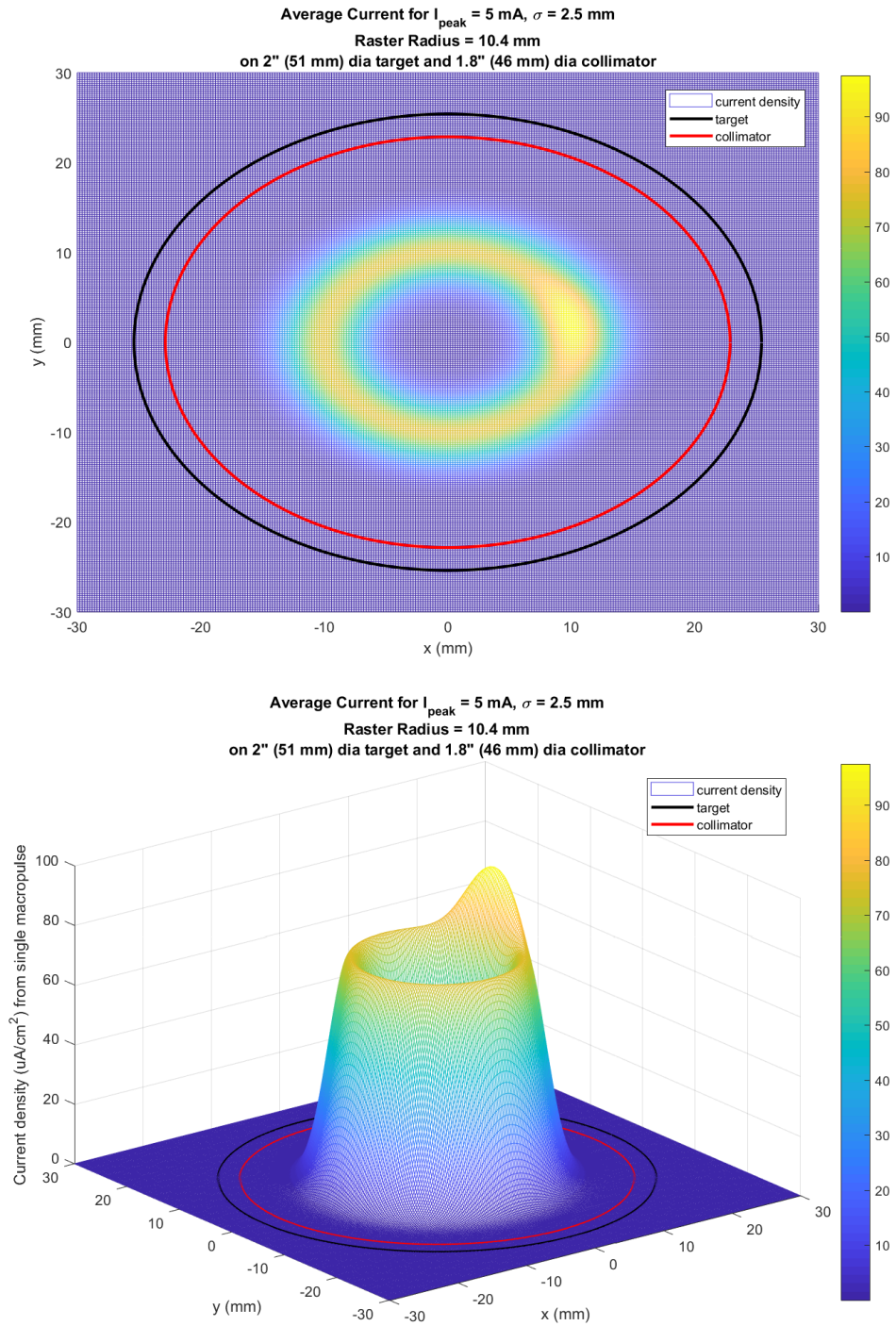


Figure 1.13: Beam Distribution from one macropulse rastered with radius 10.4 mm. Note the overlap; the beam is not perfectly radially symmetric because 3.1 circles are drawn per macropulse. The overlap occurs at different radial locations so the average beam distribution (**figure 1.10**) is more radially symmetric.

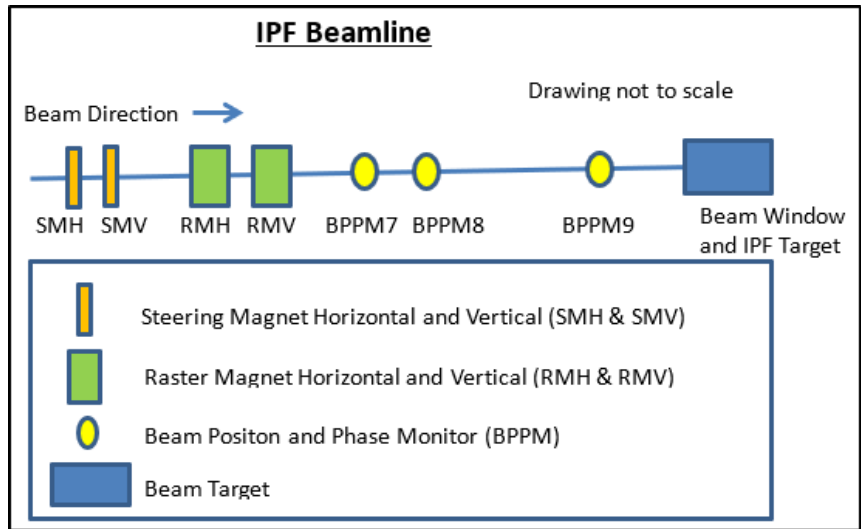


Figure 1.14: the IPF beamline includes steering and raster magnets (for both X and Y directions) and several BPPMs for measuring beam (x,y) position

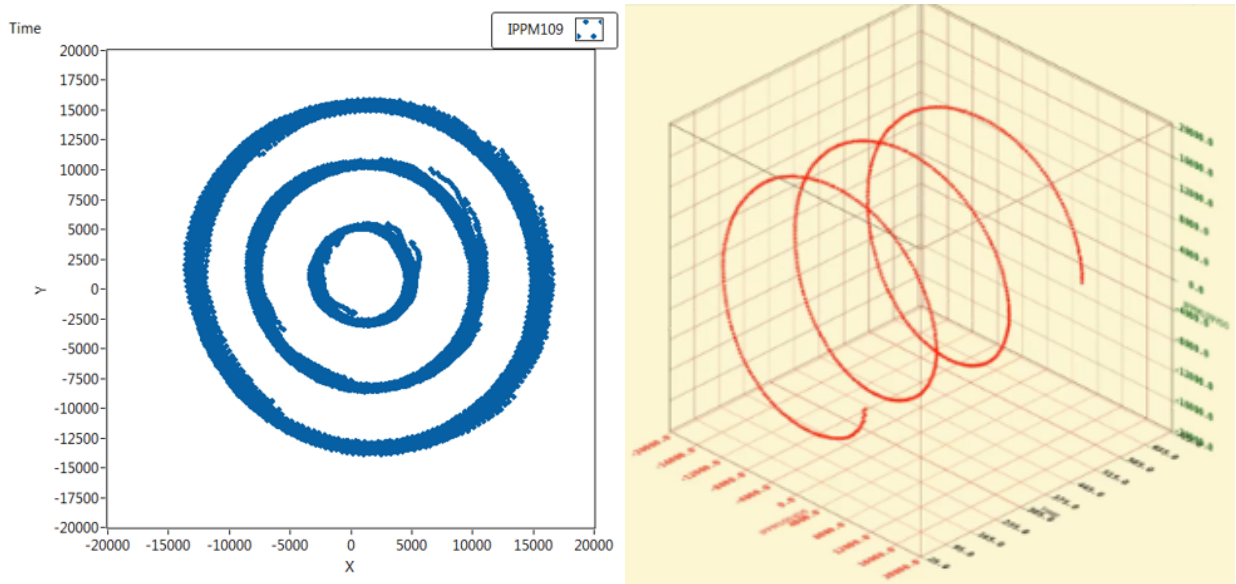


Figure 1.15: BPPMs measure beam (x,y) at 1 MHz at various z locations along the beamline. The right figure shows the time axis as well.

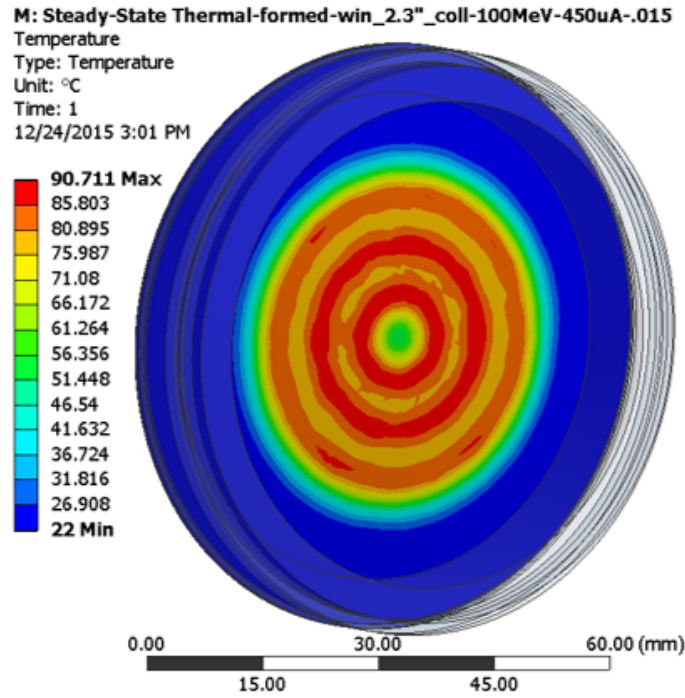


Figure 1.16: Thermal Simulation heat map for a 3 circle pattern

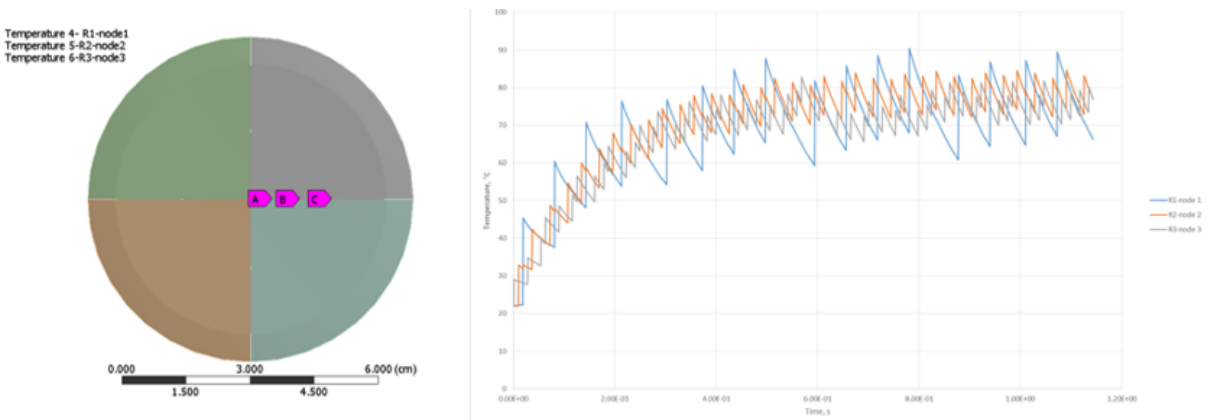


Figure 1.17: Temperature vs time for 1.2 sec, sampled at the 3 raster radii for a 3 circle pattern. Temperatures at Node A, B and C are plotted in the graph. Note that point A (closest to middle, in blue) has the hottest temperature extremes. This is because smaller raster circles deliver the same amount of beam over a smaller area, increasing the beam density. Also note that steady state is reached within about 1 second.

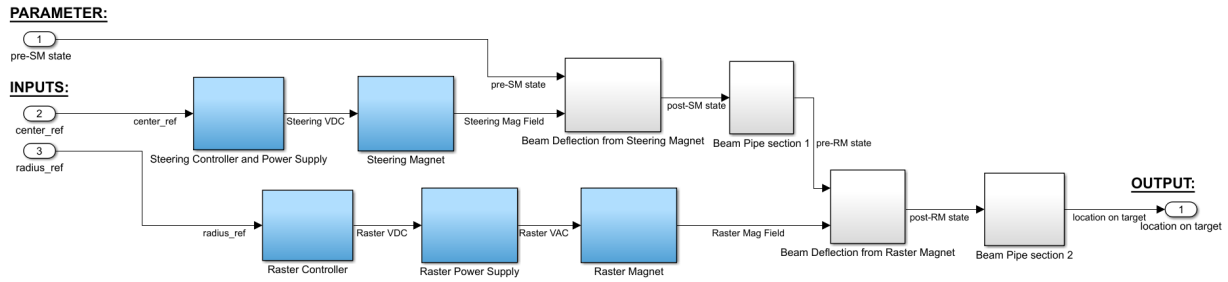


Figure 1.18: Block diagram showing components that control beam (x,y) position on target. The components with transients are blue.

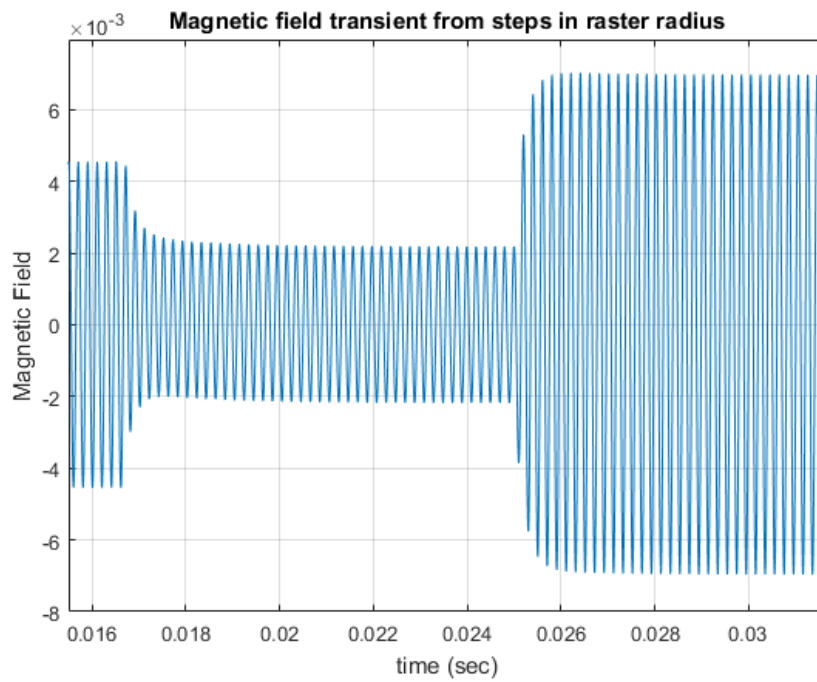


Figure 1.19: Magnetic field of raster magnets have transients when changing amplitudes. Here, the raster radius is changed around 0.017 and 0.025 sec, before which, previous transients have long decayed. This makes control of the raster system straightforward.

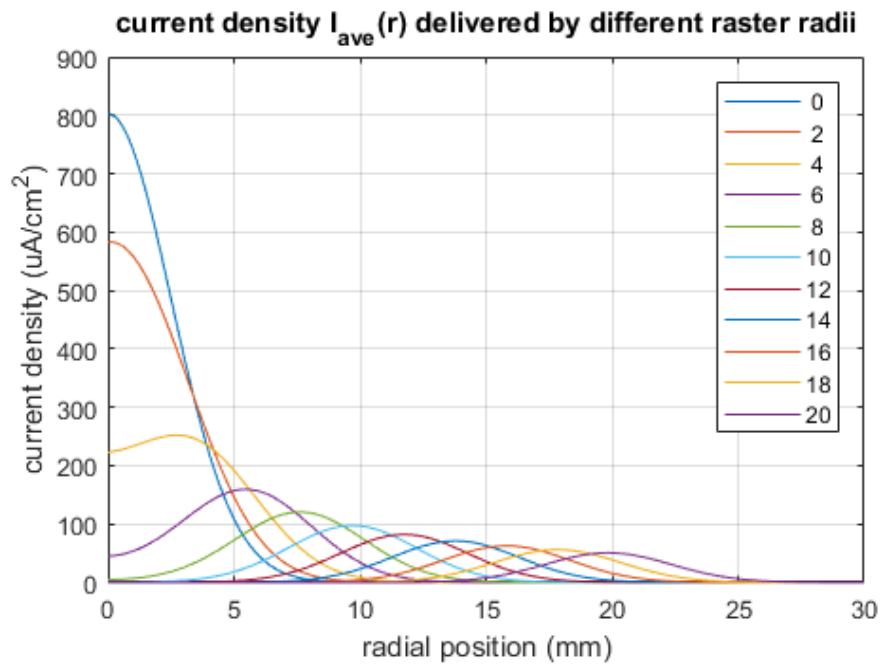


Figure 1.20: A slice of charge density delivered from a single macropulse when rastered with different radii (i.e. a slice using $y = 0$ of a graph similar to **figure 1.13**). The beam is Gaussian, so the charge density looks Gaussian far from the center at $r = 0$. For smaller r , it doesn't look Gaussian anymore because the effect of beam "spilling" from when y doesn't equal 0.

Chapter 2

Problem Scope and Formulation

The goal of this project is to improve the method for optimizing IPF beam as defined in **equations 1.4 - 1.6**, and to get specific solutions for nominal system parameters and operating conditions. However, we'll make two minor changes to simplify the problem. First, we'll change the raster frequency f from 4950 to 4800. Having $f = 4950$ results in 3.1 circles drawn per macropulse, while $f = 4800$ results in exactly 3 circles. This small change makes the problem radially symmetric to improve uniformity and simplify calculations. (4950 was used previously simply because it was the fastest frequency the power supply could operate at, and it was still approximately radially symmetric). With radial symmetry, we can replace (x, y) with r to more concisely describe the problem and simplify later calculations. With those changes, we get a new form of **equations 1.4 - 1.6**. The descriptions for each variable is restated here for convenience.

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \quad \alpha I_{\text{target}}(\theta) + \beta f(I_{\text{ave}}(r; \theta)) \quad (2.1)$$

$$\text{such that} \quad I_{\text{spill}} < I_{\text{limit}} \quad (2.2)$$

$$T(r, t; \theta) < T_{\text{limit}} \quad \forall r, t \quad (2.3)$$

with

- θ consists of parameters that describe the beam. They consist of $I_{\text{peak}}, \sigma, \mathbf{R} \in \mathbb{R}^n$, where n is the number of radii in the raster sequence before being repeated.
- $I_{\text{target}}(\theta)$ is the average total current delivered onto the target, parametrized by θ
- $I_{\text{ave}}(r; \theta)$ is the average current density delivered at radius r , parametrized by θ . This also referred to as *beam distribution* and is depicted in **figure 1.10**.
- $f(\cdot)$ is some measure of uniformity
- α, β are coefficients that weigh the relative importance of total current vs uniformity
- I_{spill} is the average total current falling on the collimator
- I_{limit} is the maximum current that the collimator can take in steady state. It is a function of the design of the collimator and its cooling system
- $T(r, t; \theta)$ is the temperature of the beam window at radius r at time t and parametrized by θ
- T_{limit} is the maximum allowable temperature for the beam window. Typically it is set as 100 C.

We define two levels of success:

1. Improve on the baseline solutions for nominal system parameters and constraint conditions.
2. Get solutions modified constraint conditions to show some degree of generalizability. In particular find good solutions for T_{limit} or I_{limit} adjusted upwards in case it is determined that we can operate the beam less conservatively.

For both, the optimization algorithm is allowed to use the baseline sequence as an initial guess since it is available.

The problem of determining the optimal distribution (i.e. the $\frac{\alpha}{\beta}$ ratio) is a question of physics, so it is not the main goal here. We're focusing on finding a method that can solve for beam parameters to produce any reasonable current distribution. In concept, there are model-free optimization methods that could bypass this issue (i.e. solve the problem from **equations 1.1 - 1.3**). These methods include Extremum Seeking and Reinforcement Learning (both discussed in Chapter 4). However, they are both iterative, and require measurements of cost J . As mentioned before, the process of measuring the isotope production rate after a run-cycle is on the scale of days or weeks, so these methods would take extremely long to test. In addition, IPF has production quotas so only very limited experimentation can be done. For these reasons we solve the problem in **equations 2.1 - 2.3**.

A bonus would be to achieve real-time feedback control of the beam distribution. This means accounting for disturbances in the system and being able to modify the raster sequence online. Major disturbances are unlikely to happen so this is not a high priority.

We now have a mathematical description of the problem and some specific measures of success. In the next chapter, we'll describe models used to describe the relationships between θ and the above quantities in the optimization problem.

Chapter 3

Models

This chapter describes how to compute the terms needed to calculate the quantities in our optimization problem we defined in **equations 2.1 - 2.3**. As mentioned earlier, going from $I_{\text{ave}}(r; \theta)$ to Isotope Production rate $P(\theta)$ is not well-modeled, so it is not attempted here. The general understanding is that more total current that is better distributed is advantageous, which is reflected in **equation 2.1**. Recall that the parameter vector θ is composed of I_{peak} , σ , and the raster sequence \mathbf{R} .

3.1 Beam distribution $I_{\text{ave}}(r; \theta)$

The beam profile model is a 2D Gaussian centered wherever the raster magnets are currently deflecting the beam.

$$I(x, y, t; I_{\text{peak}}, \sigma, \mu) = \frac{I_{\text{peak}}}{2\pi\sigma^2} \exp \left[-\frac{(x - \mu_x)^2 + (y - \mu_y)^2}{2\sigma^2} \right] \quad (3.1)$$

$$\text{with } \mu_x = r \cos(2\pi ft), \mu_y = r \sin(2\pi ft) \quad (3.2)$$

If we integrate this equation over one macropulse and look at one slice where $y = 0$ and rename x to r , we get the charge density accumulated during that macropulse (several

steps skipped)

$$dQ_{i_{\text{mp}}}(r; I_{\text{peak}}, \sigma, \mathbf{R}) = \frac{I_{\text{peak}}}{2\pi\sigma^2} \int_{t=0}^{L=625\mu\text{s}} \exp \left[-\frac{r^2 - 2r\mathbf{R}(i_{\text{mp}}) \cos(2\pi ft) + \mathbf{R}(i_{\text{mp}})}{2\sigma^2} \right] dt \quad (3.3)$$

with

- index of macropulse in sequence i_{mp}
- raster sequence $\mathbf{R} \in \mathbb{R}^n$, n =number of macropulses in a sequence

To get $I_{\text{ave}}(r)$ we simply take the sum of **equation 3.3** over the raster sequence and divide the elapsed time.

$$I_{\text{ave}}(r; I_{\text{peak}}, \sigma, \mathbf{R}) = \frac{1}{\Delta t} \sum_{i_{\text{mp}}=1}^{n_{\text{mp}}} dQ_{i_{\text{mp}}}(r; I_{\text{peak}}, \sigma, \mathbf{R}) \quad (3.4)$$

with

- number of macropulses in sequence n_{mp}
- time $\Delta t = \frac{6}{5} \frac{1}{F} n_{\text{mp}}$ (since 1 in 6 pulses is missing), with $F = 120$ Hz RepRate

Once $I_{\text{ave}}(r)$ is obtained, one can approximate I_{target} by revolving $I_{\text{ave}}(r)$ for $r < r_{\text{limit}}$ about the y-axis in order to generate a volume. Then, $I_{\text{spill}} = I_{\text{total}} - I_{\text{target}}$, where $I_{\text{total}} = I_{\text{peak}} * L * F * \frac{5}{6}$ (multiplying I_{peak} by its duty cycle).

3.2 Temperature $T(r, t; \theta)$

We simulate the temperature of the window on the water-cooled side. The temperature model assumes constant beam energy and current during the macropulse, a flat

target window with no thickness, and no radial conduction. Heat enters from the beam on one side and exits through the cooling water on the other side. It follows Newton's law of cooling with conduction coefficient τ , and heats up with specific heat α

$$\frac{\partial T(r, t)}{\partial t} = \alpha I(r, t) - \tau(T(r, t) - T_{\text{ambient}}) \quad (3.5)$$

with α, τ empirically determined based on existing temperature simulation results **figure 1.17**.

Note that the model for $I(r, t)$ is linear in I_{peak} , and the temperature model is also affine in $I(r, t)$ (linear but offset by T_{ambient}). This means that $T(r, t)$ and I_{ave} , the quantities that show up in our constraints, are affine in I_{peak} . If a solution θ violates a constraint, we can easily scale I_{peak} down to the exact value needed to meet the constraint. For example, suppose with some optimization method the solution θ' has I'_{peak} , which results in maximum temperature T_{max} . To bring T_{max} to the constraint boundary T_{limit} we compute

$$I_{\text{peak}}^* = I'_{\text{peak}} \times \frac{T_{\text{limit}} - T_{\text{ambient}}}{T_{\text{max}} - T_{\text{ambient}}} \quad (3.6)$$

with $\sigma^* = \sigma', \mathbf{R}^* = \mathbf{R}'$.

I_{spill} is linear in I_{peak} so a similar procedure can be done to bring it to the constraint boundary.

3.3 Simplifications

I_{ave} is only needed after a sufficiently long time interval, while T is needed all the time to ensure constraints are never violated. However, it will never get hotter than at the end of a macropulse because it will immediately begin cooling. We assume cooling during a macropulse is negligible since it's so short and heating typically has a much larger effect. Also, since the change in temperature is linear with current, we can add the current during a macropulse all at once, at the end of the macropulse. We can discretize the time in our system on two levels: first within the a macropulse index i_{mp} in order to get charge density delivered per macropulse $dQ_{i_{\text{mp}}}(r; I_{\text{peak}}, \sigma, \mathbf{R})$ (**equation 3.3**), and then at the end of each macropulse to get the temperature at the end of that macropulse $T_{\text{end}}(r, i_{\text{mp}})$. We also need to track the temperature at the beginning of each pulse.

To get a steady state result for temperature, we run the simulation for 100 macropulses (1 sec), after which steady state has been approximately achieved. Steady state for average current will be achieved after one whole raster sequence has been finished.

In summary, we use **equation 3.7** (time discretization of **equation 3.3**) and procedure in **equations 3.8 - 3.11**, along with averaging over the whole raster sequence (**equation 3.4**) to get $I_{\text{ave}}(r)$, and $T_{\text{end}}(r, i_{\text{mp}})$. I_{spill} and I_{target} can be obtained from $I_{\text{ave}}(r)$ as described earlier. All quantities needed to evaluate the cost and constraints are now computed.

$$dQ(r, i_{\text{mp}}; I_{\text{peak}}, \sigma, \mathbf{R}(i_{\text{mp}})) = \frac{I_{\text{peak}}}{2\pi\sigma^2} \sum_{i_t=0}^{625us/\Delta t} \exp \left[-\frac{r^2 - 2r\mathbf{R}(i_{\text{mp}}) \cos(2\pi f i_t \Delta t) + \mathbf{R}(i_{\text{mp}})}{2\sigma^2} \right] \Delta t \quad (3.7)$$

- time interval Δt

- index of macropulse in sequence i_{mp}
- raster sequence $\mathbf{R}(\mathbf{i}) \in \mathbb{R}^n, n = \text{number of macropulses in a sequence}$

$$T_{\text{beg}}(r, i_{\text{mp}} = 0) = T_{\text{ambient}} \quad \forall r \quad (3.8)$$

$$\mathbf{for} \quad i_{\text{mp}} = 0 \rightarrow \text{end} : \quad (3.9)$$

$$T_{\text{end}}(r, i_{\text{mp}}) = T_{\text{beg}}(r, i) + \alpha dQ(r, i_{\text{mp}}) \quad (3.10)$$

$$T_{\text{beg}}(r, i_{\text{mp}} + 1) = T_{\text{ambient}} + (T_{\text{end}}(r, i_{\text{mp}}) - T_{\text{ambient}}) \exp \left[\frac{t_{\text{beg}}(i_{\text{mp}} + 1) - t_{\text{end}}(i_{\text{mp}})}{\tau} \right] \quad (3.11)$$

Chapter 4

Methods Overview

In chapter 2 we've defined the optimization problem with cost and constraints related to $I_{ave}(r)$ and $T(r, t)$, and in chapter 3 we've described how those 2 quantities are computed as a function of θ . In this chapter, we discuss several approaches for solving the optimization. Some of these approaches were extensively investigated and have results shown in following chapters, while others are proposed for future work.

4.1 Properties of the Optimization Problem

We first discuss some properties of the problem at hand (**equations 2.1 - 2.3**).

- Parameter vector θ consists of beam profile parameters I_{peak} and σ , as well as a raster sequence $\mathbf{R} \in \mathbb{R}^n$, where n is the number of circles in the sequence before it's repeated again. We can choose n to be smaller to reduce the dimension of the problem to make it easier to solve, but this may result in a less optimal solution. The baseline sequence is 21 long, and for our investigation we keep it approximately this long.
- This is a continuous space problem, i.e. all quantities θ , $J(\theta)$ are in \mathbb{R} or \mathbb{R}^n .

- The cost and constraints are extremely nonlinear in the parameters. It doesn't fit in the category of problems for which most structured approaches like convex optimization will work. There's no way to know that the optimum has been achieved. The cost is extremely difficult to visualize and there's no guarantees or intuition about the shape. We do notice that $I_{ave}(r)$ is independent of the order of radii in θ , but $T(r, t)$ is highly dependent on the order.

These properties make the problem very difficult to solve accurately and well. Due to its complexity, we expect that it will need a good deal of human intervention through trial and error and manual tuning. We'll favor simple approaches whose mechanism is easy to understand and visualize, so that we can guide them to good solutions.

Another issue is the question of solving this problem offline ahead of time, or online while isotopes are being produced, in a scheme where the optimization algorithm serves as a controller. Our problem is deterministic, the cost is static, and the environment is very carefully regulated to operate at the nominal specified behavior. It is well-modeled, and, in fact, simulation is required in order to compute the cost that we defined in **2.1 - 2.3** (except from target foil irradiation analysis as used in [6], but this process is only approximate and takes far too long to for use in an iterative optimization procedure). Therefore we're losing very little from simulating the cost and running the optimization offline. We will solve the problem to get θ^* , then use that as a setpoint.

4.2 Recasting the Optimization Problem, Designing a new Cost

Equations 2.1 - 2.3 best communicate the optimization intent, but it is very difficult to solve this problem. We eliminate the constraints and add them as components to the objective cost $J(\theta)$, and now solve a minimization problem defined using **equations**

4.1 - 4.8. In the figures in the results section, we refer to (the negative of) the objective of the first problem form (**equation 2.1**) as the *true cost* and **equation 4.1** as *design cost*.

$$\theta^* = \underset{\theta}{\operatorname{argmin}} J(X(\theta); \psi, \phi) \quad (4.1)$$

with

$$J(\theta) = -\alpha_1 I_{\text{target}} \quad (4.2)$$

$$-\alpha_2 \left[\sum_{r < r_{\text{limit}}} I_{\text{ave}}(r; \theta) \right] \quad (\text{area under curve}) \quad (4.3)$$

$$-\alpha_3 I_{\text{ave}}(0; \theta) \quad (4.4)$$

$$-\alpha_4 I_{\text{ave}}(r_{\text{limit}}; \theta) \quad (4.5)$$

$$+\alpha_5 \left[\max_{r < r_{\text{limit}}} (I_{\text{ave}}(r; \theta)) - I_{\text{ave}}(0; \theta) \right] \quad (\text{maximum deviation}) \quad (4.6)$$

$$+\beta_1 \left[g \left(\gamma_1 [I_{\text{spill}} - (I_{\text{limit}} - I_{\text{offset}})] \right) \right] \quad (\text{violation of aperture constraint}) \quad (4.7)$$

$$+\beta_2 \left[g \left(\gamma_2 \left[\sum_{r,t} (T(r, t; \theta) - (T_{\text{limit}} - T_{\text{offset}})) \right] \right) \right] \quad (\text{violation of temperature constraint}) \quad (4.8)$$

State	$X(r, t; \theta) = [I_{\text{ave}}(r; \theta); T(r, t; \theta)]$
Constraints	$\psi : r_{\text{limit}}, I_{\text{limit}}, T_{\text{limit}}$
Cost parameters	$\phi : \alpha_i, \beta_i, \gamma_i, r_{\text{limit}}, I_{\text{limit offset}}, T_{\text{limit offset}}$
Barrier functions	$g(\cdot)$ e.g. x^4 , for $x > 0$ or e^x

Equations 4.2 and 4.3 handle the maximal current goal, while **equations 4.4 - 4.6** handle the uniformity goal. **Equation 4.7 and 4.8** account for the constraints. Barrier functions $g(\cdot)$ are chosen to smoothly but severely increase near constraints I_{limit} or T_{limit} , and cost parameters ϕ shape these functions as well as the overall contribution of each cost component. In order to achieve meaningful and valid solutions, we'll need to play around the with cost parameters.

4.3 Solving for Cost Minimization $J(\theta)$

For solving the minimization problem **equation 4.1** directly, two simple gradient-based methods were proposed. They have been both been used extensively for other complicated high-dimensional problems. Note on terminology: both these methods are iterative methods which have parameters (like a, k, α, ω) that are used to find the optimal parameters θ ($I_{\text{peak}}, \sigma, \mathbf{R}$) for our system. The parameters of the algorithm will be referred to as *hyperparameters* to avoid confusion with parameters of our problem θ .

4.3.1 Extremum Seeking

Extremum-seeking (ES) is a model-free gradient-based method that is a very powerful tool for optimization and/or control of poorly-modeled, noisy systems, or time-varying systems. It has many variants, some of which will be discussed here. It has used extensively for both offline and real-time optimization in a huge number of industrial applications, including for tuning components of particle accelerators ([7], [8], [9]), other physics problems like laser-pulse shaping([10]), and maximizing energy generation of wind turbines and photovoltaic microconverters [11].

Concept

An introduction to ES is given in [12]. Given some cost function $J(\theta)$, the basic idea is to build a dynamic system with perturbations for θ that responds $J(\theta)$ in such a way that θ is driven towards the unknown extrema of $J(\theta)$ over time.

Full details are in [12], but the following summarizes an illustrative case of a single input system with quadratic cost. Consider the system in **figure 4.1**. The cost function under consideration is named $f(\theta)$. The integrator, multiplicative and additive perturbations are the other part of the system. Defining error as $\tilde{\theta} = \hat{\theta} - \theta^*$, we find that the *average system* follows

$$\frac{\partial \tilde{\theta}_{ave}}{\partial t} = \frac{k f'' a^2}{2} \tilde{\theta}_{ave} \quad (4.9)$$

If $k f''$ is negative, this system is stable and error goes to zero. Averaging theory guarantees that for sufficiently large ω , if the initial estimate $\hat{\theta}(0)$ is within some region of convergence of θ^* , then $\hat{\theta}$ will converge to a small neighborhood of θ^* [13]. The neighborhood (i.e. error bound) is smaller for larger ω and smaller a .

A more multivariable version (also from [12]), which we'll refer to as MV ES, is shown in **figure 4.2**, with cost $Q : \mathbb{R}^n \rightarrow \mathbb{R}$, K diagonal with elements k_i , and

$$S(t) = \begin{bmatrix} a_1 \sin(\omega_1 t) & \dots & a_n \sin(\omega_n t) \end{bmatrix}^T \quad (4.10)$$

$$M(t) = \begin{bmatrix} \frac{2}{a_1} \sin(\omega_1 t) & \dots & \frac{2}{a_n} \sin(\omega_n t) \end{bmatrix}^T \quad (4.11)$$

It would be implemented as a difference equation with a small time discretization Δt as follows:

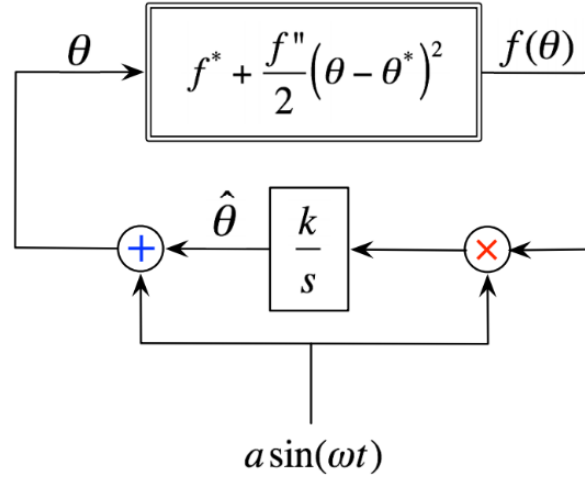


Figure 4.1: (copied from [12]) Block diagram for a simple single input cost map $f(\cdot)$. $\hat{\theta}$ is the current estimate of θ^* , and it is perturbed by signal $a \sin(\omega t)$.

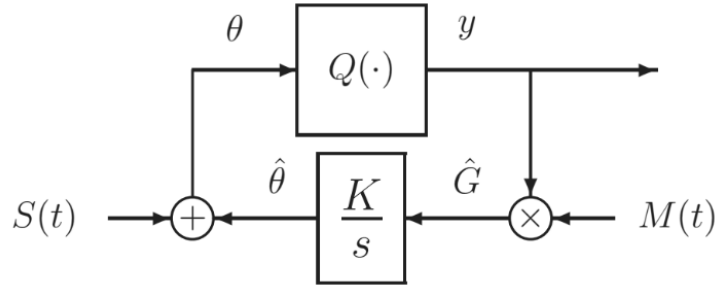


Figure 4.2: (copied from [12]) Block diagram for a multi-input cost map $Q(\cdot)$. \hat{G} is the current gradient estimate, $\hat{\theta}$ is the current estimate of θ^* , and it is perturbed by signal $S(t)$.

$$\theta_i(n+1) = \theta_i(n) + \frac{2k_i \Delta t}{a_i} \sin(\omega_i n \Delta t) J(\theta_i(n) + a_i \sin(\omega_i n \Delta t)) \quad (4.12)$$

$$\forall i, \text{ with } a, \omega, k \in \mathbb{R}^n \quad (4.13)$$

With ω_i chosen so that $\frac{\omega_i}{\omega_j}$ is rational (i.e. none are a multiple of any other) and $\omega_i + \omega_j \neq \omega_k$, if the cost map is locally quadratic, then the average system is

$$\frac{\partial \tilde{\theta}_{ave}}{\partial t} = KH\tilde{\theta}_{ave}, \text{ where H is the Hessian of the map at } \theta^* \quad (4.14)$$

If KH is negative, this system is stable. By averaging theory, for high enough ω and with initial estimate $\hat{\theta}(0)$ within some region of convergence to θ^* , $\hat{\theta}$ converges to a neighborhood around θ^* . There is also a Newton version of this algorithm whose convergence rate does not depend on the map's Hessian [12], [14]. Both of these multivariable schemes were applied and showed success in [11].

Another scheme, which we'll refer to as Bounded ES, was developed in [15]. It is based on the system in **figure 4.3**. The resulting average system follows

$$\frac{\partial \bar{\theta}_i}{\partial t} = -\frac{k_i \alpha_i^2}{2} \frac{\partial J}{\partial \theta_i}, \quad \forall i \quad (4.15)$$

This means that for the average system, the parameter vector descends the cost function in the direction of the gradient at a speed proportional to the magnitude of the gradient. As gradient approaches zero, the parameter updates approach zero. Like the previous two schemes, if the initial guess is within some region of convergence of the local minimum θ^* and sufficiently high ω is used, the system will converge within a neighborhood of θ^* .

The update equations are implemented as

$$\theta_i(n+1) = \theta_i(n) + \Delta t \sqrt{\alpha_i \omega_i} \cos(\omega_i n \Delta t + k_i J(\theta_i(n))) \quad (4.16)$$

$$\alpha, \omega, k \in \mathbb{R}^n \quad (4.17)$$

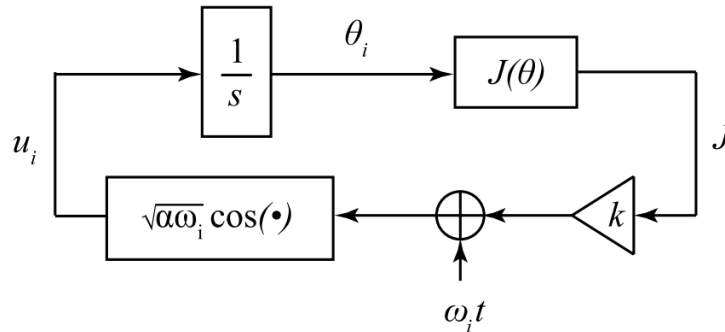


Figure 4.3: (copied from [15]) Block diagram for Bounded ES scheme for cost $J(\cdot)$. This shows the updates for a single element i of the parameter vector. The cost appearing in the cosine term ensures bounded update rates.

An illustrative 2D case is shown in **figure 4.4**. This scheme ensures a constant magnitude of update, while only the direction changes. This is useful in some applications where, when implemented as a controller on a physical system, there are limits to how fast the state can change, such as for a moving vehicle [15]. It has been successfully used on the particle accelerator at LANSCE on two occasions for beam-tuning. In [7], it simultaneously tuned a 22 element parameter vector in order to tune magnets that focus the beam in the early stage of the LANSCE accelerator. In [8], it was used to tune a 2 element vector of phase offsets that affect delivered beam intensity (also for this accelerator). In both cases, the cost function is very complex and poorly modeled due to hysteresis of magnets and factors such as temperature causing the lengthening or shortening wires, which causes uncertain system delays.

Note the difference in average systems for MV ES and Bounded ES (**equations 4.14,4.15**). For stability of MV ES, the sign of K should be opposite of the sign of H . For stability of the Bounded ES system, the sign of k_i should be positive if searching for a minimum and negative if searching for a maximum. The fact that the sign of this hyperparameter doesn't depend on the cost J is advantageous if J is complicated and poorly understood (as in our case).

There are schemes which work for general nonlinear systems with internal dynamics,

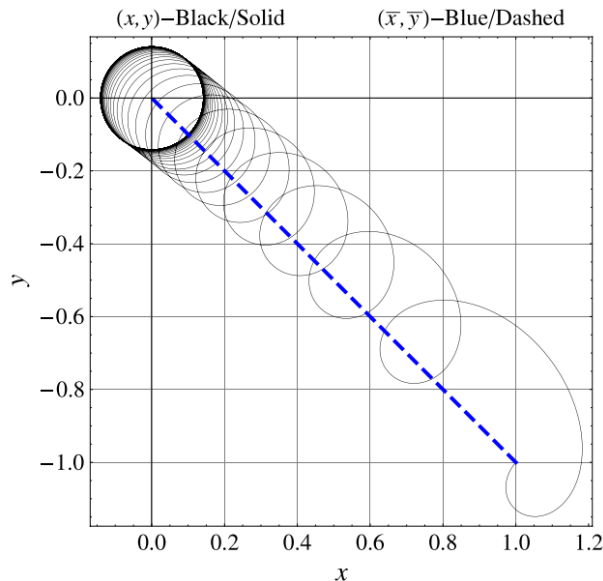


Figure 4.4: (copied from [15]) Illustration of Bounded ES for a 2D parameter vector in with quadratic cost. While the magnitude of update is constant, the turning rate is higher when moving in the wrong direction, and slower when in the correct direction. On average, the estimate approached the minimum at $(0, 0)$.

such as the one analyzed in [16]. These are interpreted as online implementations which has ES acting as an adaptive controller, such as [9], [10]. In that case, the cost is a function of state x , which has dynamics $\dot{x} = f(x, \alpha(x, \theta))$, where parameters θ parametrize control inputs to that system. It is in this case that ES is notably powerful as an adaptive controller. When ES is applied to our optimization problem, $J(\theta)$ is a static map where θ is simply an argument to the cost function rather than a control input to a dynamic system. Therefore, we restrict our discussion to the schemes illustrated above which are for static maps.

Several other schemes have been proposed and used, including versions with stochastic rather than periodic noise ([17]) and discrete-time versions with both periodic ([18] and used in [9], [10]) and stochastic ([19]) perturbations. Stochastic perturbations are useful in cases where periodic perturbations are undesirable or impossible, such as in certain biological processes. This is not an issue here. However, the randomness of perturbation may be preferred to the periodic perturbation in the MV ES case in high-dimensional

systems (the MV ES cases relies on orthogonality of between elements, i.e., ω_i should have a good spread without any harmonics overlapping). In general, these versions require similar assumptions on the system and give similar guarantees to convergence within a neighborhood of a local minima.

Summary of ES

Many ES methods exist which are proven to converge to a neighborhood of a local extrema when the initial guess is within some region of convergence, if the parameters of the ES system are chosen wisely. This means ω must be high enough (for the true system to match the average system) and k must be of the correct sign (for stability of the average system in MV ES and to travel in the correct direction for Bounded ES). The hyperparameter a (or α in the Bounded ES case) is another parameter that affects convergence rate and the size of the neighborhood in which convergence is guaranteed. This method only requires computation of the cost $J(\theta)$ at each step, and the algorithm does not need knowledge of the analytical form of $J(\theta)$ or its gradient.

Since our cost function has may have many unknown local minima, and the region of convergence for each one is qualitative rather than exactly known, no form of ES guarantees a global solution from an arbitrary initial guess. However it may be useful for finding local solutions that are still significantly better than the baseline.

4.3.2 Gradient Descent

Gradient Descent is a model-based method for optimization, used in a huge range of applications, perhaps most notably for for training artificial neural networks for classification and regression problems [20]. Its applications also include finding parameters for adaptive filters like LMS [21]. An application that doesn't involve fitting model parameters to data is error-minimization for GPS algorithms [22], but many more exist for other physical

problems.

Concept

The goal of Gradient Descent, like ES, is to find the minimum of a cost function. However, in this case the analytic form of the the cost function in terms of the parameters to be optimized is assumed to be known.

We'll refer to the most basic form of gradient descent as Steepest Descent. Starting at an initial guess $\hat{\theta}(0)$, it evaluates the gradient at that point, $\nabla_{\theta}J(\theta)$. Then it takes a step in that direction, with magnitude scaled by a parameter α . The update equation is

$$\theta_{k+1} = \theta_k - \alpha(\nabla_{\theta}J(\theta_k)), \quad \alpha \in \mathbb{R}^n \quad (4.18)$$

For J with Hessian H with maximum eigenvalue λ_{\max} , if $\alpha < \frac{2}{\lambda_{\max}}$, Gradient Descent will converge within a neighborhood of stationary point where $\nabla_{\theta}J = 0$ [21]. Once the neighborhood of the minimum is reached, one may reduce α to shrink the neighborhood of convergence. The region of convergence for this minimum is the “bowl” around the minimum, up until where $\nabla_{\theta}J \neq 0$.

One issue of Steepest Descent is convergence rate, for example when the cost function has highly elongated level sets. Only using 1st order information at only θ_k may not be enough. This may be solved by Newton's method, which also uses the Hessian H to determine the step. This works well if the cost function is quadratic or near quadratic.

$$\theta_{k+1} = \theta_k + \alpha[H(\theta_k)]^{-1}(\nabla_{\theta}J(\theta_k)), \quad H \in \mathbb{R}^{n \times n}, \quad \text{typically } \alpha = 1 \quad (4.19)$$

Other more computationally efficient ways to speed up learning without resorting to

computing the Hessian and its inverse include adaptively changing components of step size α [23], [24]. For our problem, convergence to the most optimal point is the main concern rather than speed so those methods aren't employed.

A bigger concern for us is being trapped in local minima. One possible remedy for this (which also may speed up convergence) is using “momentum” [20], as implemented below. The size of the step depends on both the gradient and the size of the previous step.

$$\theta_{k+1} = \theta_k - (\alpha(\nabla_{\theta}J)_k + \beta\Delta\theta_k), \quad \alpha, \beta \in \mathbb{R}^n \quad (4.20)$$

Others variants of Gradient Descent used for fitting parameters to data in statistical models include Stochastic Gradient Descent and Batch Gradient Descent [20]. Those are useful when loss is defined with respect to a model fitting error (like Mean Squared Error), where there are multiple samples of observations X and outputs Y for a single set of parameter values θ_k that can be used to compute the objective. Here, only one J is defined by one θ , so those methods are not applicable for solving this problem as defined in **equation 4.1**.

4.3.3 Summary: ES versus Gradient Descent

Both methods have been proven useful in many applications for optimization of very complex, high dimensional cost functions. ES is model-free and requires design of several parameters a, k, ω to find the optimum, while Gradient Descent requires α (and possibly β if using momentum). Gradient Descent evaluates the exact gradient of the cost function to make a step. Both methods guarantee convergence to a neighborhood of local minima if the initial guess is within some region of convergence. The exact region of convergence is not quantitatively known for ES, but is well understood for Gradient Descent.

4.4 Alternate Approaches

Two other methods were considered for solving **equation 4.1**: using genetic algorithms, and casting the problem as a optimal control problem and solving it with a reinforcement learning method. Genetic Algorithms were not attempted, but they are briefly discussed here. A particular reinforcement learning method was attempted and the problem setup is discussed here as well.

4.4.1 Genetic Algorithms

Genetic Algorithms are non-gradient based methods for solving optimization and search problems [25]. They have been particularly useful in discrete computational problems such as scheduling systems [26], but have also been useful in engineering optimization problems, such as design of turbine blades in a jet engine [27].

A genetic algorithm is based on the evolutionary processes of natural selection, crossover of traits, and mutations. Rather than tracking a single candidate solution, it keeps track of a population of solutions in each step. To initialize the algorithm, a group of candidate solutions is encoded in some representation. During each iteration, each candidate is evaluated for “fitness” to the objective. A subset of them are selected for the next iteration. Then crossover step occurs, which blends traits from this subset to create successors. Occasionally, mutations are introduced for increased diversity. This group of successors is used in next iteration, and iterations continue until performance goals are reached or changes are too small.

Compared to our methods above, genetic algorithms have some advantages. They are model-free and don’t make any assumptions about the cost function other than that an optimum exists. The cost function can be extremely non-smooth. Constraints are easily handled by simply not allowing them to exist in the population. Since there is population

rather than a single point, there is less chance being trapped in local minima, and the parameter space can be more completely explored. If initialized well and the process of crossover and mutation are done wisely, it can find a fairly global optimum.

The main disadvantages compared to ES and Gradient Descent are in its complexity to understand, implement, and tune. It is inherently a stochastic method and there are no convergence guarantees to even local minima. The choice of selection and crossover criteria depend strongly on the specific problem and require good domain knowledge. For these reasons this method was not pursued.

4.4.2 Casting as an MDP, solving with Reinforcement Learning

For either the true cost $J(\theta)$ or curve-fitting loss $l(\theta)$, we can directly solve the respective minimization problems for $\theta^* = [I_{\text{peak}}^*, \sigma^*, \mathbf{R} \in \mathbb{R}^n]$, with n being the number of radii in the raster sequence before it repeats again. This is an extremely high dimensional space if we want n to be large, which would allow for more optimal solutions. The baseline sequence used right now has $n = 21$.

An alternative is to notice that we could frame the problem as a discrete time *Markov Decision Process (MDP)* whose accumulated stage cost over infinite time is either true cost $J(\theta)$. An MDP requires a state space, action space, stage cost, and transition model that takes x_t to x_{t+1} . We can consider I_{peak}, σ as part of the transition model and define each action at each time t to be the next raster radius $R \in \mathbb{R}$. Then solving the MDP cost minimization will result in a policy $\pi(x)$ that maps from each state $X(r) = [I_{\text{ave}}(r), T(r)]$ to a radius R to draw next. The MDP cost minimization problem is more fully defined in **equation 4.21** [28], and solving it means solving for $\pi(x) \quad \forall x \in \mathcal{X}$. To retrieve raster sequence, we start at an initial state and run the policy, recording the actions that it took.

$$J^* = \min \quad J^\pi(x) = \left[\sum_{t=0}^{\infty} \gamma_t g(x_t, \pi(x_t)) \mid x_0 = x \right] \quad \forall x \in \mathcal{X} \quad (4.21)$$

with

- $x_t \in \mathcal{X}$ (states, i.e. average current and temperature at beginning of next pulse
 $X(r) = [I_{\text{ave}}(r); T_{\text{beg}}(r)]$)
- $\pi(x_t) : \mathcal{X} \rightarrow \mathcal{U}$ policy mapping states to actions, i.e. raster radius $R \in \mathbb{R}$
- $g(x, u) : \text{stage cost at each state } x \text{ and performing action } u$
- $\gamma : \text{discount factor (not the same } \gamma \text{ as in eqs 4.7,4.8, and usually close to 1)}$

Dynamic Programming would produce a globally optimal solution. However, we notice that our problem has a continuous state space $X(r)$ and continuous action space \mathbb{R} , while Dynamic Programming methods like Policy or Value Iteration work for discrete state and action spaces. In fact, the state space is actually infinite dimensional since $I_{\text{ave}}(r)$ and $T_{\text{beg}}(r)$ are functions. We would select a discretization for r to define the dimensions of our state space, and discretizations for values that each of these states could take. For this problem to be remotely accurate, the memory and computational requirements would be far too high for Dynamic Programming to be feasible. A better approach would be methods for continuous state and action spaces. Suitable Reinforcement Learning methods are Policy Gradient methods such as DDPG, REINFORCE for continuous state and action spaces [29], [30]. Those two methods use artificial neural networks to represent the policy π . They have shown success in high dimensional state and action spaces simulation environments such as *OpenAIGym* and *Mujoco* [31], [32]. We would still need to define discretization for radial positions r , and our state would be $[I_{\text{ave}}(r), T_{\text{beg}}(r)]$ at all of these positions.

Here we have a much smaller “action space” to search over, \mathbb{R} instead of \mathbb{R}^n . This could potentially be an easier problem to solve. Also, we would arrive at a policy rather than a single answer, allowing for closed-loop control.

However, there’s now an enormous state space that was not relevant before. We would need to solve this for all the relevant states that we may visit. It’s not immediately clear whether solving an MDP will be easier than solving the original problem. Other shortcomings are:

- Parameters in the optimization: Neither I_{peak} nor σ can be solved in the optimization. One can run the algorithm multiple times with different I_{peak}, σ combinations to get optimal \mathbf{R}^* , then compare them afterwards.
- Accuracy: $I_{\text{ave}}(r)$ would need to be tracked as a filter instead of a moving window average in order to preserve the Markov property, so this would introduce some modeling error.
- Reward/Cost shaping: We need to define the cost/reward per state that produce the desired behavior. It may be more difficult to shape these functions than the ones listed before in **equation 4.1**. We also need to define our initial states and length of episode (i.e. time-horizon) appropriately.
- Convergence: Solving this problem using DDPG or REINFORCE involves training a neural network to represent a policy which maps states to actions. Inherently there is no convergence guarantee here. It depends on many design choices such as network architecture, learning rate for the networks, and batch size for training. It will be harder to evaluate performance and tune these hyperparameters.
- Policy Initialization: ES, Gradient Descent, and genetic algorithms have a straightforward way to start the optimization algorithm with a good guess. Here, we would need

to initialize an entire policy function. *Imitation Learning* methods like GAIL [33] exist to initialize policies, but this is a whole task in itself. Without initialization, convergence is even more difficult.

Using REINFORCE was attempted (chosen over DDPG to ensure the solution was unbiased), but due to the difficulties above it was not further pursued.

4.5 An Easier Sub-Problem

The problem defined in **equations 2.1 - 2.3** and reformulated in **equation 4.1** is very difficult because of the complex constraint and cost functions. In particular, the temperature constraint is very difficult since it strongly depends on the order of circles drawn.

With our knowledge about the problem, we can devise a multi-step approach. First, we can use a structured optimization approach to find a raster sequence \mathbf{R} and beam width σ for the objective of achieving a desired “unit height” beam distribution $I_{\text{ave}}(r)$, without considering the temperature. Then, we can manually reorder the raster sequence through trial and error for minimal temperature extremes. This would mean placing the small circles evenly spaced as possible within the sequence, since they deposit the highest beam power density and produce the most severe temperature variations. Then we would increase I_{peak} until either the constraints on I_{spill} or T_{limit} is violated.

This approach would not have an optimality guarantee since we are prescribing the desired beam distribution. It would need some more trial and error for post-processing. It could still end up with better results, since it is an easier cost to solve for the structured optimization algorithm. Another benefit is that cost-shaping is easier: we determine a desired shape such as a uniform distribution (**figure 1.4**), rather than need to tune the weights of many different cost parameters in **equations 4.2 - 4.8**.

There are two proposed methods for achieving a specified beam distribution $I_{\text{ave}}(r)$: Gradient Descent and Expectation Maximization. They were not extensively investigated but could be the subject of future work.

4.5.1 Gradient Descent

Another way we can use Gradient Descent is to minimize the a “curve-fitting” loss such as squared error, instead of the one in **equation 4.1**.

$$l(\theta) = \sum_{r_i}^N (X(r_i, \theta) - X^*(r_i))^2 \quad (4.22)$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \quad l(\theta) \quad (4.23)$$

where $X(r; \theta) = I_{\text{ave}}(r; \theta)$, r_i are different sample points, and N is the number of sample points. Note that since each θ can generate different datasets to lead to different $l(\theta)$, we can use techniques like Stochastic Gradient Descent and its variants which may have better convergence properties. It’s not clear whether this cost is easier to minimize than the original one.

4.5.2 Expectation Maximization

Expectation Maximization (EM) [34] is a statistical algorithm for Maximum Likelihood Estimation (MLE) for parameters of a statistical model when the model depends on unobserved variables. Broadly speaking, it alternates between an expectation (E) step where the expected log-likelihood is computed, followed by a maximization (M) step which finds the parameters that maximize this expectation.

A popular use case is to learn the parameters of a *mixture model* for a probability

distribution, which is a mixture of multiple components, each with a different distribution and relative weight. In the case of a mixture of Gaussians, the E and M steps have closed forms so that implementation is fairly straightforward [35]. The parameters include the mean, covariance, and relative weight of each Gaussian. The user only needs to determine the number of components to use and how to initialize them. The algorithm finds the parameters that best match the observed dataset.

For our problem, we can generate a dataset whose probability distribution has the same shape as the desired $I_{\text{ave}}(r)$. The components we have available for matching the dataset would be in the form of **equation 3.3**, where $\mathbf{R}(i_{\text{mp}}), i_{\text{mp}} = 1, 2, \dots, k$ are the parameters being learned for k components. The E and M steps would be derived analytically or done numerically.

Alternately, we can try in 2D. Generate a desired distribution $I_{\text{ave}}(x, y)$ to be represented as the mixture of many Gaussians. Use the same E and M steps as for a Gaussian mixture model, except that σ must be the same across all components. Once the algorithm converges, inspect the locations of the Gaussians $\mu_i = (x_i, y_i)$ to figure out which radii to choose and construct a raster sequence from those results. A downside of this method is that the learned parameters have 2 degrees of freedom when they really should have 1 (the radius from center). We'd need to hope that the results have μ arranged in concentric circles so that it can be achieved by our raster system.

4.6 Summary

We desire a method to solve **equation 4.1** as optimally as possible. The optimization will be done offline since the problem is well-modeled and the system is well-controlled with few disturbances. Since the problem has such high dimension with such a complex objective function, we expect that finding a global solution will be very difficult. Our approach will

need lots of fine tuning and human intervention, so we desire simple algorithms that are predictable and straightforward to tune. ES and Gradient Descent are the two classes of algorithms considered. Other methods were discussed, but they do not make up the main results.

Also note that since this will be computed offline rather than implemented as an online controller, after a simulation of either of these methods with certain hyperparameters, we have access to the entire trajectory of the optimization. We can take the parameter values wherever in the optimization that is optimal, rather than only at the end. Our goal is to pass through the lowest minima we can, without requiring that it is where we end up in the end. Using momentum Gradient Descent and Extremum Seeking will ensure that plenty of exploration will be done in search of these minima.

Chapter 5

Methods Used

This chapter describes in better detail how methods were used to solve this specific problem.

5.1 Implementation

Gradient Descent and ES for directly minimizing design cost $J(\theta)$ are the two methods that were extensively investigated. As mentioned before, all optimization was done offline through simulation. Python was chosen to implement these algorithms and compute all quantities needed. In particular, Gradient Descent required computing gradients at every step. Since we're using a model and simulating the system, we have the analytic form of the gradient. It is very difficult to write out by hand, and it would change with different discretizations of the problem. *PyTorch*, a Python package geared for machine learning applications, was used to compute gradients more easily. It tracks operations done on the parameter vector θ in order to compute J , so that the gradient can be easily computed. Another Python package or symbolic math tool with similar functionality could be used instead.

5.2 Procedure

There were two main challenges in getting meaningful results. First, one must choose the cost hyperparameters for a cost that, when minimized, result in a desired beam distribution that satisfies the constraints. This means determining the trade off between uniformity and total current delivered, as well as determining the shape of barrier functions to prevent violation of constraints. The second challenge was to find hyperparameters for the particular method that succeed at minimizing the cost that's chosen. This usually meant reducing the sensitivity to the cost to ensure smaller steps were taken. The goals of correct cost-shaping and convergence were often at odds. Intuitive cost-shaping would require very severe barrier functions. This would make the cost landscape more extreme so that convergence is more difficult. A structured procedure was used to achieve results:

1. Define the goal: determine the desired shape for $I_{\text{ave}}(r)$ (i.e. prioritization of maximum current vs maximum uniformity).
2. Define inputs: cost parameters, hyperparameters, and an initial guess.
3. Run the optimization to achieve lowest design cost. During each step, save the design cost and its components. Also, compute the true cost (**equation 2.1**) of the associated valid solution, obtained by scaling I_{peak} according to **equation 3.6** in order to not violate constraints.
4. At the end of the optimization, select the parameters for the valid solution with lowest true cost, and inspect the resulting total current, uniformity, and $I_{\text{ave}}(r)$ (**figure 5.2**). If they are good enough, the process is complete.
5. If results need improvement, inspect the graphs for $I_{\text{ave}}(r)$ and $T(r, t)$, as well as the optimization trajectory. Determine whether to adjust hyperparameters for cost shaping or convergence. For example, if after scaling, the temperature is at the limit

and I_{spill} isn't near I_{limit} , then we know that temperature was the limiting quantity. The cost should increase the penalty of temperature to move it away from that limit and utilize the “budget” on I_{spill} .

As an example, the graphs and tables in **figures 5.1 - 5.5, table 5.1** were produced for one particular trial with Gradient Descent. Also note it was decided to use raster sequences of length 20 instead of 21. This would keep our initial guess close to the baseline. Also, since it's a multiple of 5, the 1/6 missing pulses will always occurs in the same place in the sequence for consistency.

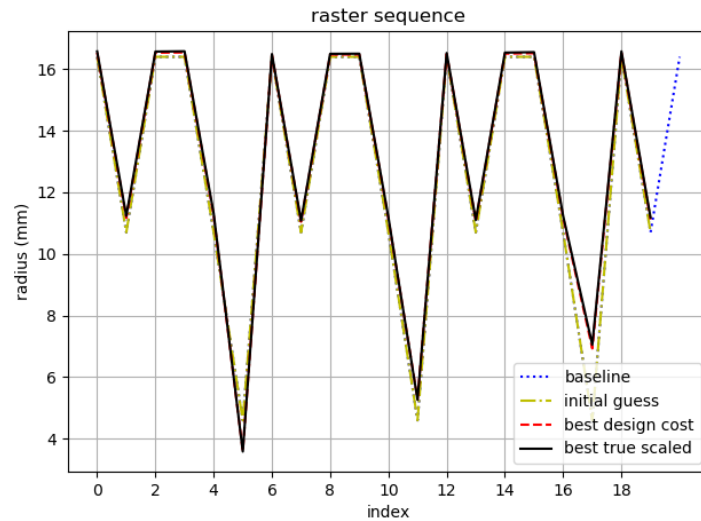


Figure 5.1: Raster sequence determined from one trial of Gradient Descent. For comparison, it also shows the baseline and initial guess used for this run. *Best design cost* in is shown in red but is covered by *best true cost*. Notice that the sequence hasn't changed much, other than the small circles at the end being slightly larger.

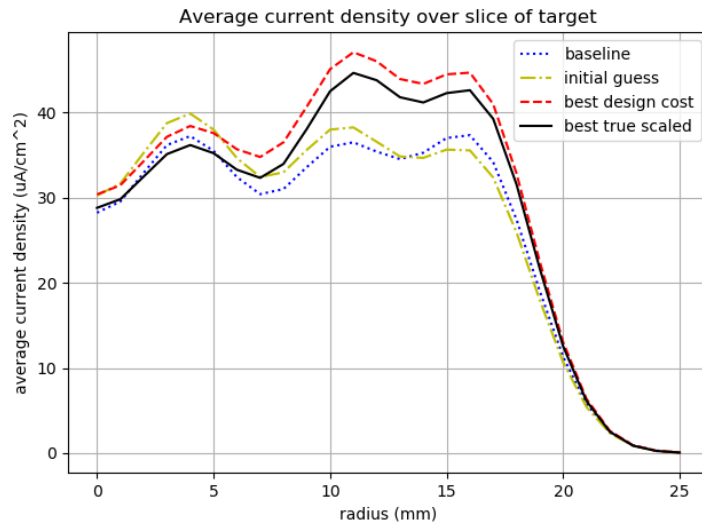


Figure 5.2: $I_{ave}(r)$ at the end of the trial. The red dashed line shows the result that had the lowest design cost, an invalid solution because of a constraint violation. The black line shows the best valid solution that was ever encountered during the optimization. Recall that I_{target} , the amount of current in uA delivered, is the volume obtained by rotating this graph around the y axis. Here we see that there's more total current for our solution than in the baseline, without reducing coverage at the center of the target ($r = 0$).

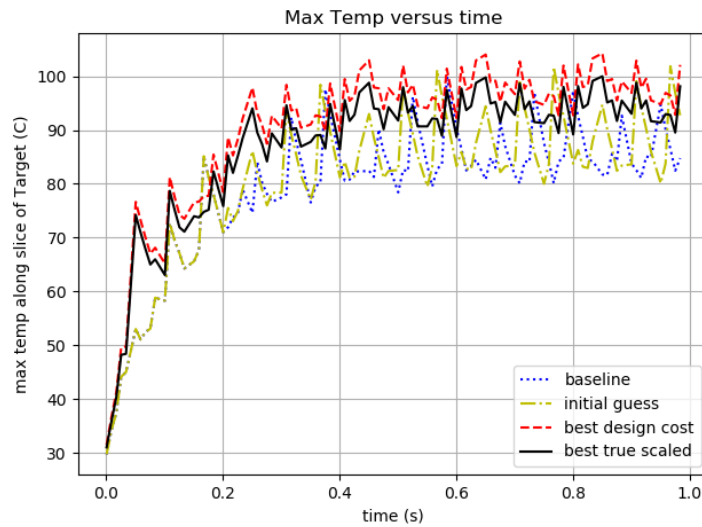


Figure 5.3: The hottest temperature on the target as a function of time. The points plotted are actually only measured at the end (hottest part) of each macropulse. We see that despite having much more current (as shown in **figure 5.2**), our solution (in black) does not violate the temperature constraint of 100 C.

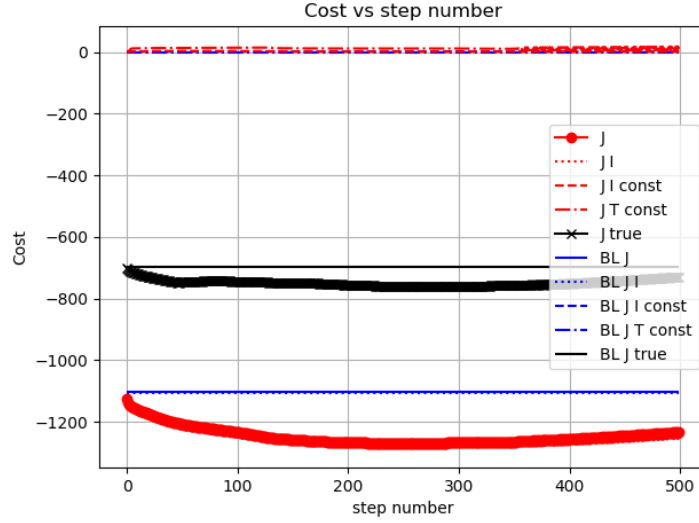


Figure 5.4: The cost trajectories as a function of iteration number during the algorithm. Total design cost (J) is plotting along with the component due to current, I_{limit} constraint, T_{limit} constraint, and true cost. The corresponding components for the baseline parameters are plotted for comparison. In this graph, we see that both design (thick red) and true cost (thick black) drift lower than their corresponding baselines, indicating good convergence behavior. At the end they drift slightly back up. This may be from a hard constraint on parameters, preventing gradient descent from stepping in the correct direction.

Table 5.1: Table of outputs to determine how successful this run was. The final result is the last row. We see that this solution produced had higher I_{peak} and put more current on the target without violating the constraints (1.9 uA and 100 C). T_{max} is at the constraint, and is the limiter for I_{peak} , while I_{spill} still has some margin. We may get a better solution by increasing the penalty of temperature so that a “cooler” solution is found, and I_{peak} can be scaled higher for more total current.

Solution Name	I_{limit}	T_{limit}	I_{peak}	σ	I_{target}	$I_{\text{ave}}(r=0)/I_{\text{ave,max}}$ (%)	I_{spill}	T_{max}
baseline	1.9	100	6.7	2.5	416.8496	75.61074626	1.9003906	100.057
initial guess	1.9	100	6.7	2.5	416.95618	75.85303164	1.7938232	101.864
best design cost	1.9	100	8.13715	2.45627	506.6608	60.45279571	1.911377	104.479
best true scaled	1.9	100	7.72918	2.45456	481.2517	59.92927537	1.8217773	100

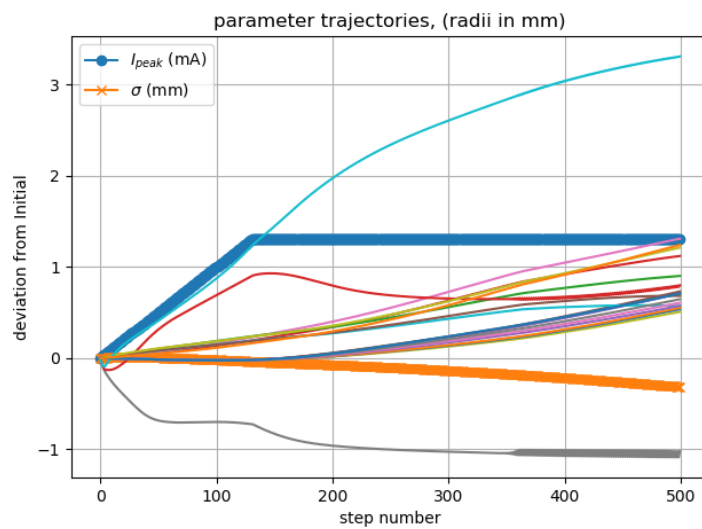


Figure 5.5: Parameter trajectories as a function of iteration number. I_{peak} and σ have markers, while the other curves are raster radii in the sequence. Here we see that most parameters changed smoothly, but I_{peak} stopped abruptly when it ran into a hard boundary that was defined for it.

Chapter 6

Results

This chapter illustrates the results obtained using specific methods described in the previous chapter.

As a reminder, the goals were to find, for different trade-offs of total current and uniformity, improved solutions for both the nominal system and one with different constraint values.

6.1 Comparison of Methods

Of the methods discussed in chapter 4, Bounded ES and plain Gradient Descent were the ones that yielded good results. MV ES and another discrete-time version in [18], using the design guidelines in described in [9], were attempted. Despite extensive tuning of the hyperparameters, both of these methods very quickly diverged within a few steps. The cost and high dimension of this problem proved too difficult for these methods, which require the function to be locally quadratic to guarantee local convergence. Newton-based ES wasn't investigated since the standard MV ES version didn't show promise. Bounded ES did not have these instability issues since it had bounded updates by its nature. REINFORCE was attempted but was not successful because of difficulties with initialization, and not further

pursued due to the expected complexity, accuracy, and cost/reward-shaping challenges.

The cost functions were found to be quite difficult to optimize. In an earlier version of the problem, the I_{spill} condition was mistakenly enforced as a limit on $I_{\text{ave}}(r_{\text{limit}})$ rather than on total current outside r_{limit} . In this case, Gradient Descent and ES required the same amount of tuning to achieve useful results. When the constraint form was corrected (to better reflect the actual failure mode of the collimator), Gradient Descent would always quickly diverge, even with much reduced step size α . It is suspected that this change significantly modified the shape of the cost function, since it now depended on $I_{\text{ave}}(r)$ in a more complicated way. A gradient clipping step (not a new concept) which limited the 2-norm of each step fixed this issue and allowed it to be still effective. Bounded ES still usually converged without modification. Momentum Gradient Descent did not show improvement over plain Gradient Descent so results with that method are not shown.

6.2 Results for Bounded ES and Gradient Descent

Since the other methods did not yield relevant results, we show some detailed results only for Bounded ES and Gradient Descent (with gradient clipping) for the 3 goals stated above. In general, Gradient Descent performed better than Bounded ES. It tended to be easier to tune because there were fewer parameters, and it usually reached more optimal solutions.

6.2.1 Nominal system parameters, for different goals

Three different goals were defined: maximum current, maximum uniformity, and the “mixed” goal which would unambiguously improve on the baseline in both areas. This means depositing more current overall, without reduce coverage in the middle of the target.

For both methods, it was typically easiest to solve the maximum current goal, and

it was very difficult to beat the baseline in terms of uniformity. This might be from the fact that more intense temperature spikes are a direct result from putting more beam in the middle, making the cost function very difficult there. Also, the baseline sequence is already quite uniform, so it was hard to beat. A more uniform distribution could not be found at all with Bounded ES. Gradient Descent found a solution using the older constraint formulation mentioned above (figure 6.5). Table 6.1 shows a summary of results., and figures 6.1 - 6.4 show the associated raster sequences and beam distributions.

Table 6.1: Summary of results for ES and Gradient Descent (GD) for nominal constraints. All solutions increased I_{peak} in order to achieve higher I_{target} without violating the constraints. The measure of uniformity shown is $\frac{I_{\text{ave}}(0)}{\max_r I_{\text{ave}}(r)}$. For this metric, the baseline couldn't be beat. Gradient Descent proved to be more effective at increasing I_{target} .

Solution Name	Constraints		Beam Profile		Performance Metrics		Constraint metrics	
	I_{limit} (uA)	T_{limit} (C)	I_{peak} (mA)	σ (mm)	I_{target} (uA)	$I_{\text{ave}}(r=0)/I_{\text{ave,max}}$ (%)	I_{spill} (uA)	T_{max} (C)
Baseline	1.9	100	6.70	2.50	416.80	75.60	1.90	100.00
ES: max current	1.9	100	7.51	2.52	467.30	44.53	1.86	100.00
GD: max current	1.9	100	7.97	2.30	496.03	33.88	1.89	100.00
ES: mixed	1.9	100	7.45	2.62	463.94	62.40	1.88	100.00
GD: mixed	1.9	100	7.63	2.44	475.07	62.36	1.81	100.00

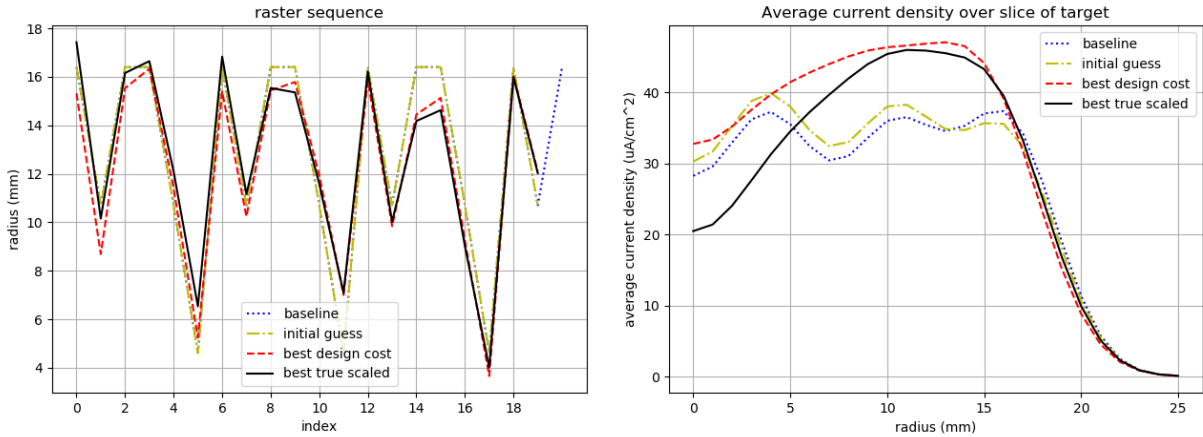


Figure 6.1: Solution obtained from Bounded ES for **maximum current** goal. The first graph shows the raster sequence. Notice that this shows more deviation from the baseline than for the Gradient Decent example in figure 5.1. This was always the case. The second graph shows the resulting beam distribution. As expected, it gives up coverage on the center of the target for more total current.

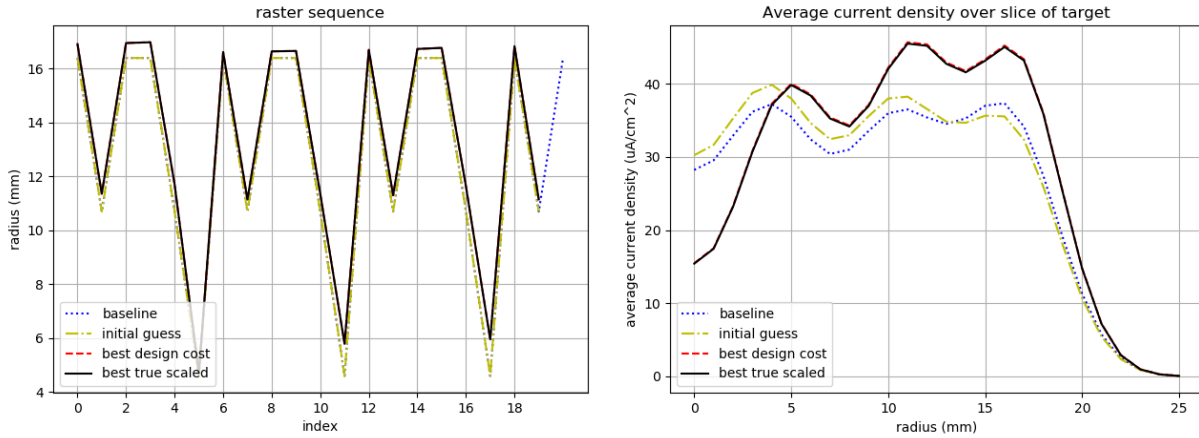


Figure 6.2: Solution obtained from Gradient Descent for **maximum current** goal. The circles are slightly larger so that I_{peak} can be increased without overheating the center. There is higher current density at larger radial position r which greatly increases total beam delivery.

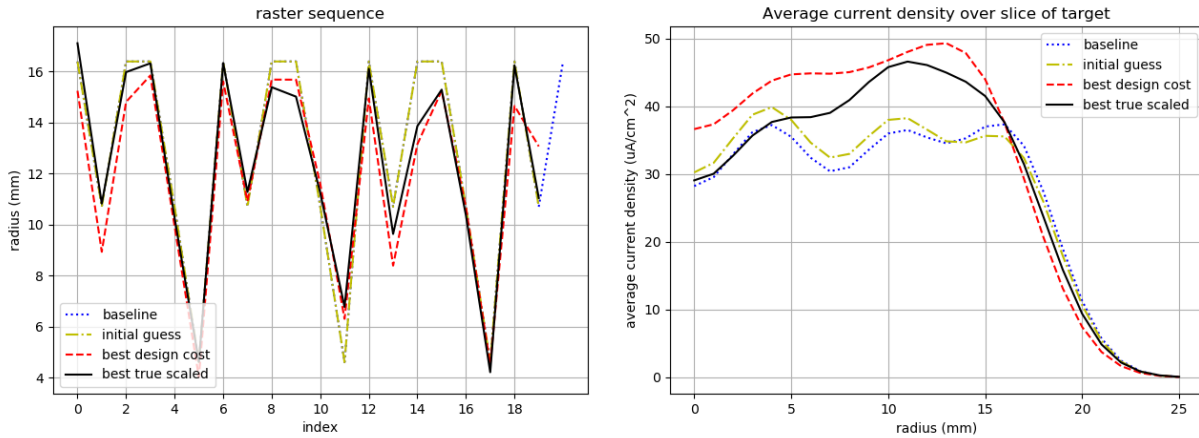


Figure 6.3: Solution obtained from Bounded ES for **mixed** goal. Coverage at the center of the target wasn't compromised in achieving more overall beam delivery.

6.2.2 Off-nominal Constraint Values

We consider cases for higher T_{limit} and higher I_{limit} . This would be useful in the cases that it is determined that isotope production can be run less conservatively, or the cooling systems are upgraded or qualified for higher performance. Both methods were still able to find good solutions. The results for the mixed goal are shown for each method, and summary of their performance are in **table 6.2**. **Figures 6.6 - 6.11** show the associated

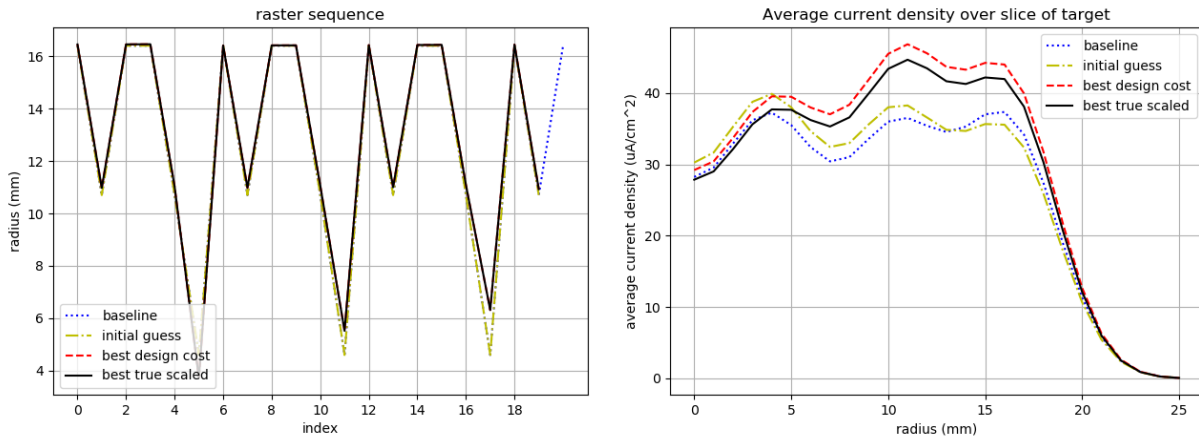


Figure 6.4: Solution obtained from Gradient Descent for **mixed** goal. As in **figure 6.3**, $I_{ave}(0)$ wasn't reduced when achieving higher I_{target} .

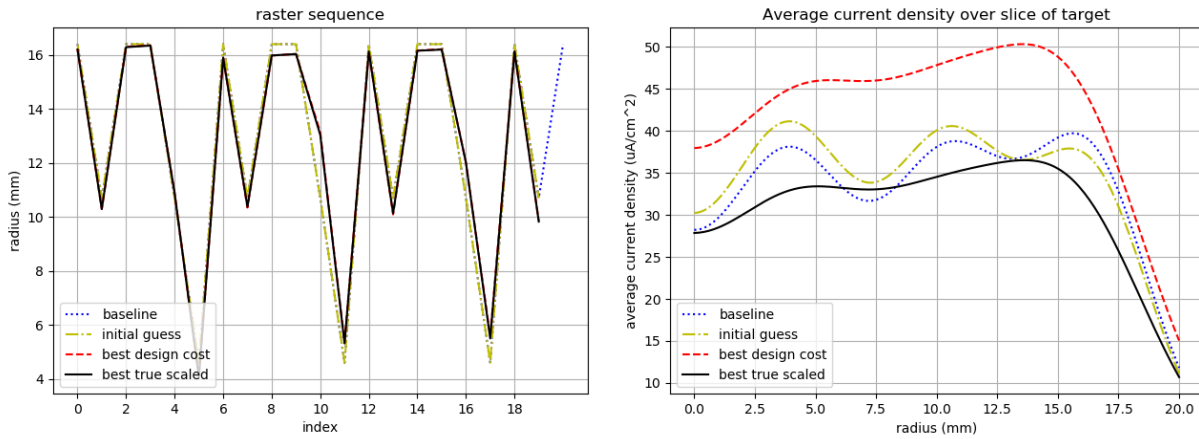


Figure 6.5: Solution obtained from Gradient Descent on a system with simpler constraint on for I_{spill} . This cost function was easier to solve for this particular goal. As expected, this trades off a great deal of current delivery for uniformity. In practice, total beam delivery is more important than uniformity so other solutions would be more likely to be used.

raster sequences and beam distributions.

Table 6.2: Summary of results for ES and Gradient Descent (GD) for modified constraints. The three cases are higher I_{limit} , then T_{limit} , and then higher both. For baseline, I_{peak} is scaled up while σ and raster sequence are held constant, until one of the constraints is violated. This only results in change for the 3rd case. We again see that Gradient Descent gave the best solutions for total beam delivery and the baseline had the best percentage uniformity.

Solution Name	Constraints		Beam Profile		Performance Metrics		Constraint metrics	
	I_{limit} (uA)	T_{limit} (C)	I_{peak} (mA)	σ (mm)	I_{target} (uA)	$I_{\text{ave}}(r=0)/I_{\text{ave,max}}$ (%)	I_{spill} (uA)	T_{max} (C)
Baseline	2	100	6.70	2.50	416.80	75.60	1.90	100.00
ES	2	100	7.50	2.65	466.85	65.48	1.99	100.00
GD	2	100	7.68	2.48	477.87	63.60	1.98	100.00
Baseline	1.9	120	6.70	2.50	416.80	75.60	1.90	100.00
ES	1.9	120	8.99	2.72	560.07	66.71	1.86	120.00
GD	1.9	120	9.75	2.24	607.83	51.47	1.85	120.00
Baseline	2	120	7.05	2.50	438.62	75.61	1.99	104.13
ES	2	120	8.95	2.75	557.36	68.36	1.99	120.00
GD	2	120	9.75	2.28	607.60	53.50	1.95	120.00

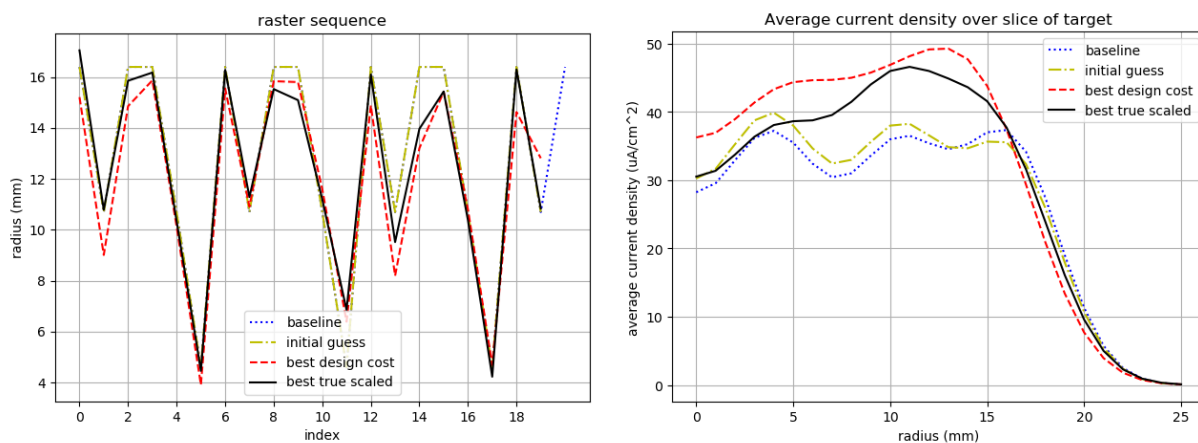


Figure 6.6: Solution obtained from Bounded ES for mixed goal for system with increased I_{limit} .

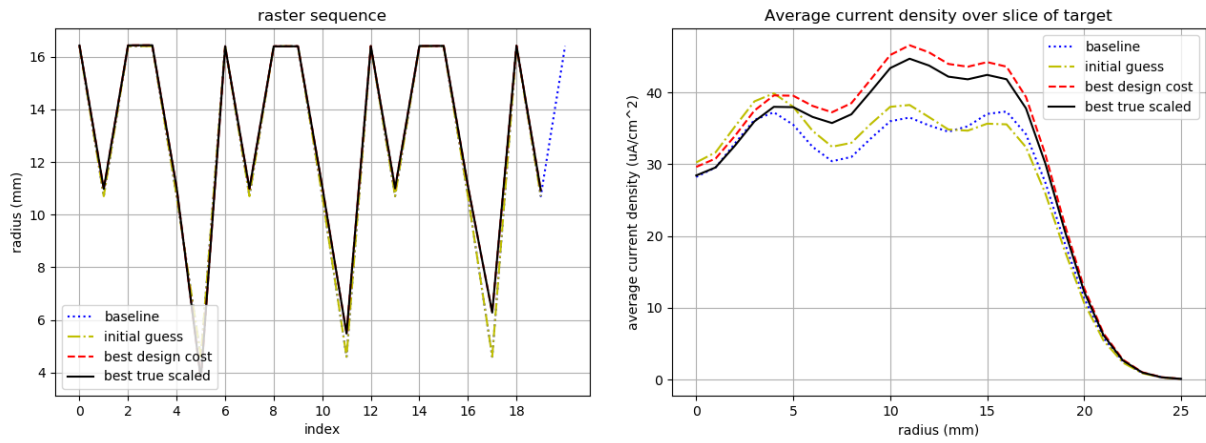


Figure 6.7: Solution obtained from Gradient Descent for mixed goal for system with increased I_{limit} .

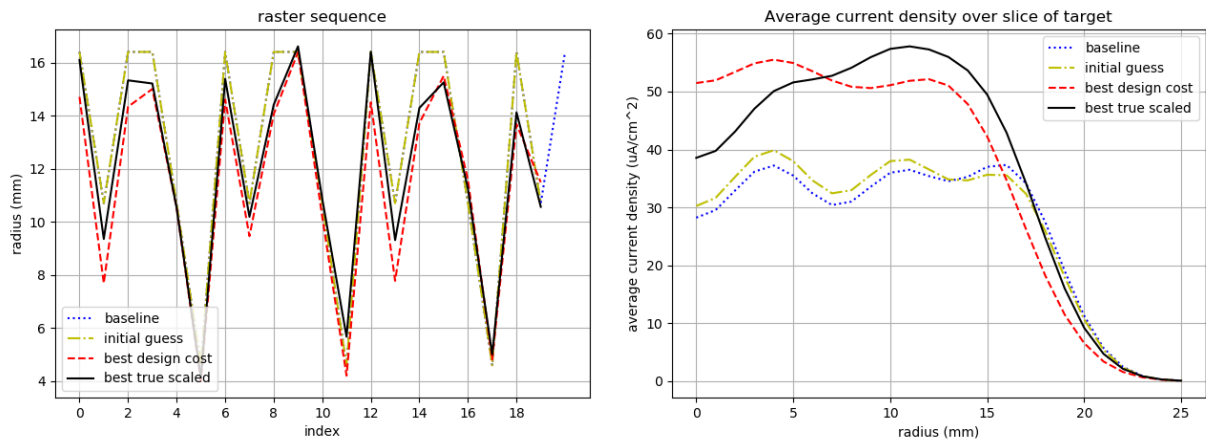


Figure 6.8: Solution obtained from Bounded ES for mixed goal for system with increased T_{limit} .

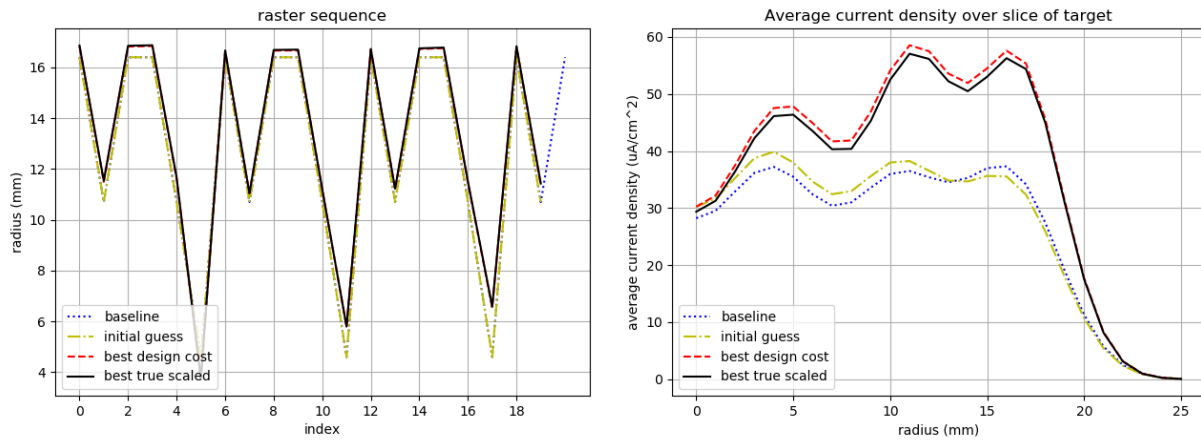


Figure 6.9: Solution obtained from Gradient Descent for mixed goal for system with increased T_{limit} .

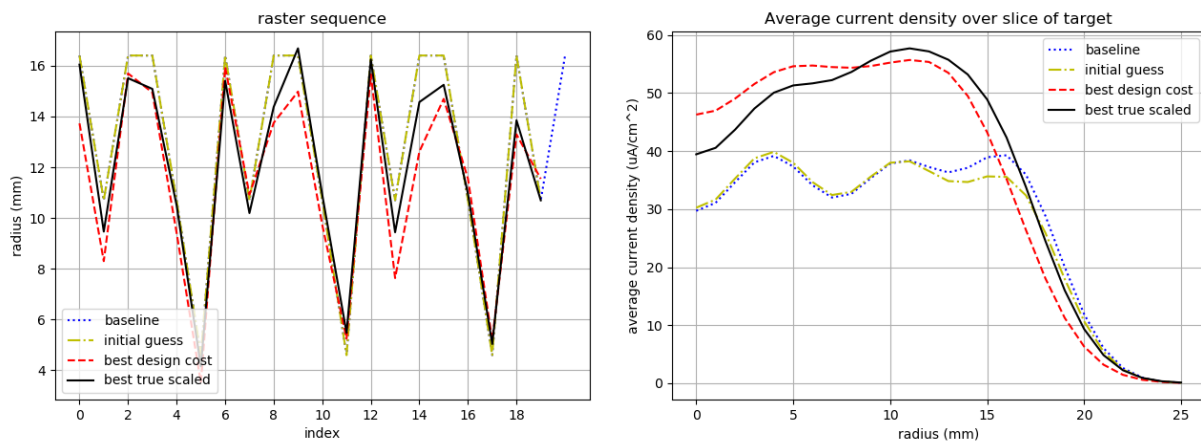


Figure 6.10: Solution obtained from Bounded ES for mixed goal for system with both limits increased.

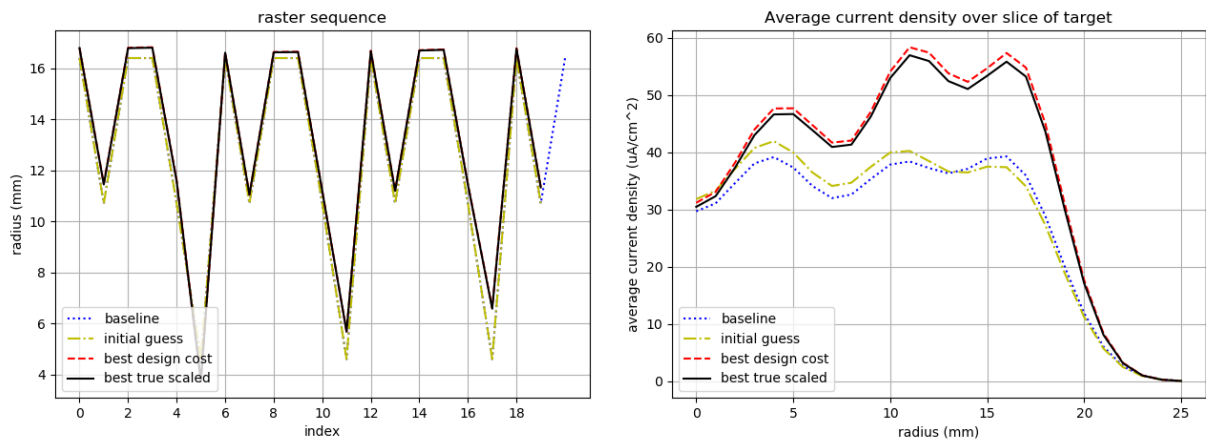


Figure 6.11: Solution obtained from Gradient Descent for mixed goal for system with both limits increased.

Chapter 7

Conclusions and Future Work

In this document, we introduced the problem of optimizing beam for isotope production. We then explained the models developed for simulating the system and discussed several general approaches to solving the problem. This involved placing the constraints in the original objective function to arrive at **equations 4.1 - 4.8**. Extremum Seeking and Gradient Descent were the two methods that were extensively investigated for solving this optimization because of their relative simplicity which were helpful for tuning for a problem so complex.

We've found that using our family of cost functions in **equations 4.1 - 4.8**, Bounded ES and Gradient Descent (with gradient clipping) were effective at exploring the parameter space in the neighborhood of the baseline. Aided with the procedure described in Chapter 5 which exploited the linearity of the model with respect to parameter I_{peak} , we were able to improve the baseline significantly. In the process, we found that Gradient Descent tended to do better and was easier to tune than Bounded ES due to it having fewer hyperparameters.

Further work could be done to improve the method used. The challenge was shaping the cost and finding hyperparameters for the algorithm that would lead to a useful solution. It was admittedly a difficult task for some goals. Choosing a better cost function family

that better communicates design intent with better convergence properties may make this method easier to use.

Other approaches discussed in Chapter 4 could be tried to get better results. The MDP framework and associated reinforcement learning algorithms such as DDPG and REINFORCE have shown effectiveness in other continuous space, high dimensional problems and might be effective here as well. Formulating an easier sub-problem as described in chapter 4.5 may be an easier method to use.

Even with the results so far, the impact can be huge. IPF's isotopes serve roughly 30,000 patients a month in medical imaging applications. Therefore even small percentage increase in isotope yield have a significant impact. As seen in **table 6.1**, one solution increased current delivery to from 416 to 475 μA without sacrificing any beam coverage on the center of the target. This 14 percent increase in total current will translate to a similar increase in number patients that IPF can serve.

Granted, the problem formulation used is not complete. As mentioned in chapter 1, there are a few other failure modes resulting from coupled physical mechanisms occurring in the targets that cannot be modeled. Other parts of the system can be optimized such as the geometry of the target assembly and the water cooling system. IPF is pushing frontiers globally in terms of beam delivered on target, and this effort is only one part of a wider effort to increase isotope production.

Bibliography

- [1] A. Koehler, R. Schneider, and A. Sisterson, “Flattening of proton dose distributions for large-field radiotherapy,” *Medical Physics*, vol. 4, no. 4, 1977.
- [2] G. Steyn, C. Vermeulen, A. Botha, J. Conradie, J. Crafford, J. Delsink, J. Dietrich, H. du Plessis, D. Fourie, Z. Kormany, M. van Niekerk, P. Rohwer, N. Stodart, and J. de Villiers, “A vertical-beam target station and high-power targetry for the cyclotron production of radionuclides with medium energy protons,” *Nuclear Instruments and Methods in Physics Research A*, vol. 131, no. 144, pp. 136–138, 2013.
- [3] V. Bechthold and R. Nickles, “Target foils,” in *1st International Workshop on Targetry and Target Chemistry*, pp. 5–8, 1985.
- [4] J. Kolski, “Proposed varying amplitude raster pattern to uniformly cover target for the Isotope Production Facility at LANSCE,” in *Instrumentation and Beam Material Interactions*, pp. 148–150, 2014.
- [5] Y. Batygin, “Uniform target irradiation by circular beam sweeping,” *Nuclear Instruments and Methods in Physics Research A*, vol. 363, pp. 128–130, 1995.
- [6] J. Kolski, J. Audia, H. Bach, Y. Batygin, J. Bradley, M. Connors, J. Engle, E. Espinoza, E. Figueroa, E. Hall, M. Martinez, M. Nortier, D. Reass, W. Roybal, and H. Watkins, “Varying amplitude raster pattern for high power isotope production targets,” in *6th International Particle Accelerator Conference*, pp. 2298–2300, 2015.
- [7] A. Scheinker, “Model independent beam tuning,” in *Proceedings of the 2013 International Particle Accelerator Conference*, pp. 1862–1864, 2013.
- [8] A. Scheinker, S. Baily, D. Young, J. Kolski, and M. Prokop, “In-hardware demonstration of model-independent adaptive tuning of noisy systems with arbitrary phase drift,” *Nuclear Instruments and Methods in Physics Research A*, vol. 756, pp. 30–38, 2014.
- [9] E. Schuster, C. Xu, N. Torres, E. Morinaga, C. Allen, and M. Krstic, “Beam matching adaptive control via extremum seeking,” *Nuclear Instruments and Methods in Physics Research A*, vol. 581, pp. 799–815, 2007.

- [10] B. Ren, P. Frihauf, R. J. Rafac, and M. Krstic, “Laser pulse shaping via extremum seeking,” *Control Engineering Practice*, vol. 20, pp. 674–683, 2012.
- [11] A. Ghaffari, M. Krstic, and S. Seshagiri, “Extremum seeking for wind and solar energy applications,” *Mechanical Engineering*, vol. 136, 2014.
- [12] M. Krstic, *Extremum Seeking Control*, pp. 1–5. London: Springer London, 2013.
- [13] H. Khalil, *Nonlinear Systems*. Pearson Education, Prentice Hall, 2002.
- [14] A. Ghaffari, M. Krstic, and D. Nesic, “Multivariable newton-based extremum seeking,” *Automatica*, vol. 48, pp. 1759–1767, 2012.
- [15] A. Scheinker and M. Krstic, “Extremum seeking with bounded update rates,” *Systems and Control Letters*, vol. 63, pp. 25–31, 2014.
- [16] M. Krstic and H.-H. Wang, “Stability of extremum seeking feedback for general nonlinear dynamic systems,” *Automatica*, vol. 36, pp. 595–601, 2000.
- [17] S.-J. Liu and M. Krstic, “Stochastic averaging in continuous time and its applications to extremum seeking,” *IEEE Transactions on Automatic Control*, vol. 55, pp. 2235–2250, 2010.
- [18] J.-Y. Choi, M. Krstic, K. B. Ariyur, and J. S. Lee, “Extremum seeking control for discrete-time systems,” *IEEE Transactions on Automatic Control*, vol. 47, pp. 318–323, Feb 2002.
- [19] C. Manzie and M. Krstic, “Discrete time extremum seeking using stochastic perturbations,” in *IEEE Conference on Decision and Control*, pp. 3096–3101, 2007.
- [20] S. Ruder, “An overview of gradient descent optimization algorithms,” *CoRR*, vol. abs/1609.04747, 2016.
- [21] S. Haykin, *Adaptive Filter Theory (3rd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.
- [22] A. Chiablaem, P. Supnithi, S. Klinngam, C. Panachart, A. Saekow, P. Kenpankho, K. Watthanasangmechai, T. Yokoyama, T. Tsugawa, and M. Ishii, “Estimation of the single gps-receiver bias using the gradient descent algorithm,” in *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 1–5, 2016.
- [23] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.
- [24] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.

- [25] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1989.
- [26] D. Corne, H.-L. Fang, and C. Mellish, “Solving the module exam scheduling problem with genetic algorithms,” *Proceedings of the Sixth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, 1993.
- [27] J. H. Holland, “Genetic algorithms,” *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.
- [28] N. Atanasov, “Infinite horizon problems and stochastic shortest path.” University Lecture Notes, 2018.
- [29] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *CoRR*, vol. abs/1509.02971, 2015.
- [30] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine Learning*, vol. 8, pp. 229–256, May 1992.
- [31] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym,” *CoRR*, vol. abs/1606.01540, 2016.
- [32] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, Oct 2012.
- [33] J. Ho and S. Ermon, “Generative adversarial imitation learning,” *CoRR*, vol. abs/1606.03476, 2016.
- [34] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 39, no. 1, pp. 1–38, 1977.
- [35] N. Vasconcelos, “Expectation maximization.” University Lecture Notes, 2018.