

Expedited Real Time Processing for the NETL HYPER Cyber-Physical System

Jesus Arias, Jalyn Gordon, Dr. Comas Haynes*, Dr. Aklilu Giorges*
 Georgia Institute of Technology, *Georgia Tech Research Institute



Project Overview

- Cyberphysical systems, specifically hardware-in-the-loop simulations, are becoming more attractive due to the subsequent opportunity to test traditionally cost-prohibitive equipment through a wide range of implementations and conditions.
- Challenges include the integrated system time difference arising from the physical system dynamics, response times of sensors, the computational model and computational time, control systems, and overall systems interactions.



The National Energy Technology Laboratory's (NETL) Hybrid Performance (HyPer) facility is a cyberphysical system comprised of:

- Real Physical Components
 - Compressor, Turbine, Load Bank, Generator, Post Combustor, and Air Plenum
- Simulated Components
 - Solid Oxide Fuel Cell System Model, Fuel Valve Model, and Gasifier Model

These work together in real time to simulate the coupled behavior of this design for the development of effective control schemes. Figure 1 provides a photograph of the actual facility. Figure 2 below illustrates the schematic of this system.

Figure 1: Overhead photograph of HyPer facility.

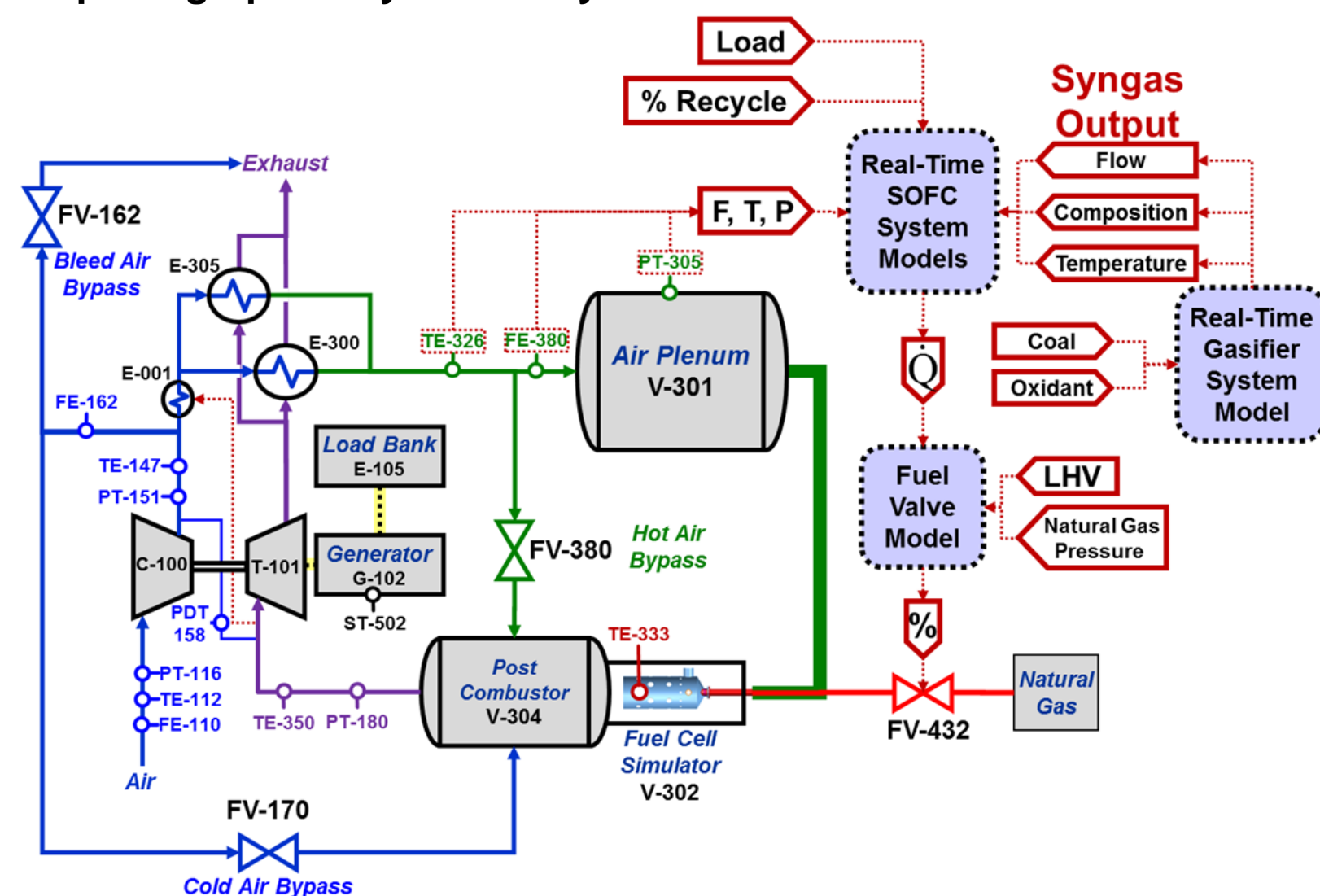


Figure 2: Schematic of NETL HyPer Facility. The computational components are labeled with dashed lines and red connections and include the Real-Time SOFC System Model, Fuel Valve Model, and Real-Time Gasifier Model. The primary object of interest for this project is the SOFC model.

Problem Statement

In the case of HyPer, issues arose from differing time between the response of the hardware gas turbine system and the computationally simulated 1-D solid oxide fuel cell (SOFC) subsystem. Measures are proposed in order to decrease the simulation time to 5 msec (down from 40 msec) for a 5 msec real-time simulation. The overall project presents three proposed solutions involving the following:

- Optimization of the existing SOFC model
- Offline "a priori" simulations of key parameters
- Alternative numerical methods to the current Crank-Nicolson solution scheme

Current time auditing efforts have identified the functions that serve to calculate electrochemical potentials as a primary culprit in terms of iterations and self-time.

The present focus of this study is to investigate item (1), whether the convergence criteria for these functions are too narrow, or if the iterative solver must be replaced with a faster algorithm.

Methodology

Initial time auditing revealed:

- Electrochemical algorithm is a highly iterative process
- Electrochemical algorithm took majority of calculation time
- Must evaluate if convergence criteria for these functions are too narrow, or if the iterative solver must be replaced with a faster algorithm.

Figure 3 illustrates a flow diagram of the iterative process.

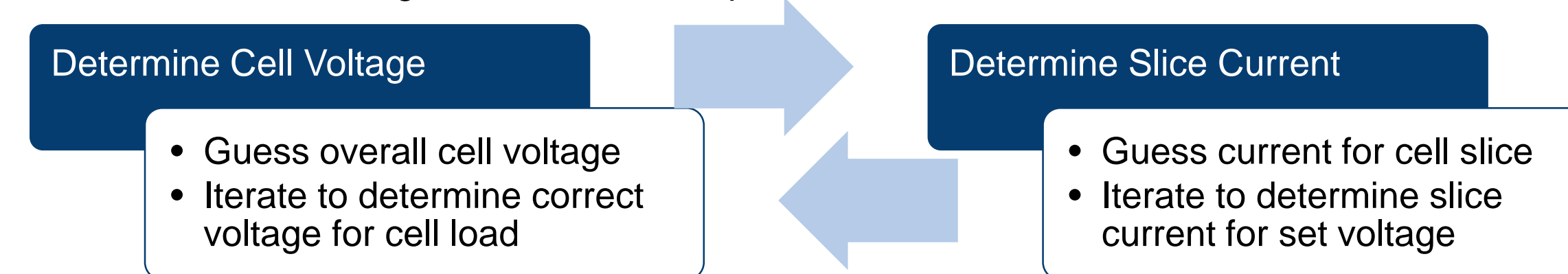


Figure 3: Flowchart of iterative process for determining Voltage-Current relationship for fuel cell. Both use rootfinding recipes to determine values and subsequently must be solved simultaneously for a given timestep.

The electrochemical algorithm is defined by two rootfinding processes defined below as Equations (1) and (2) and taking into account cell reactant composition.

$$I(V) - I_{load} = 0 \quad (1)$$

$$V(I) - V_{guess} = 0 \quad (2)$$

The original scheme uses the bisection method to solve equation (1) and the false position method to solve (2). Four different rootfinding methods were tested on a representative problem to illustrate the possible benefits. The convergence properties are illustrated in Figure 4.

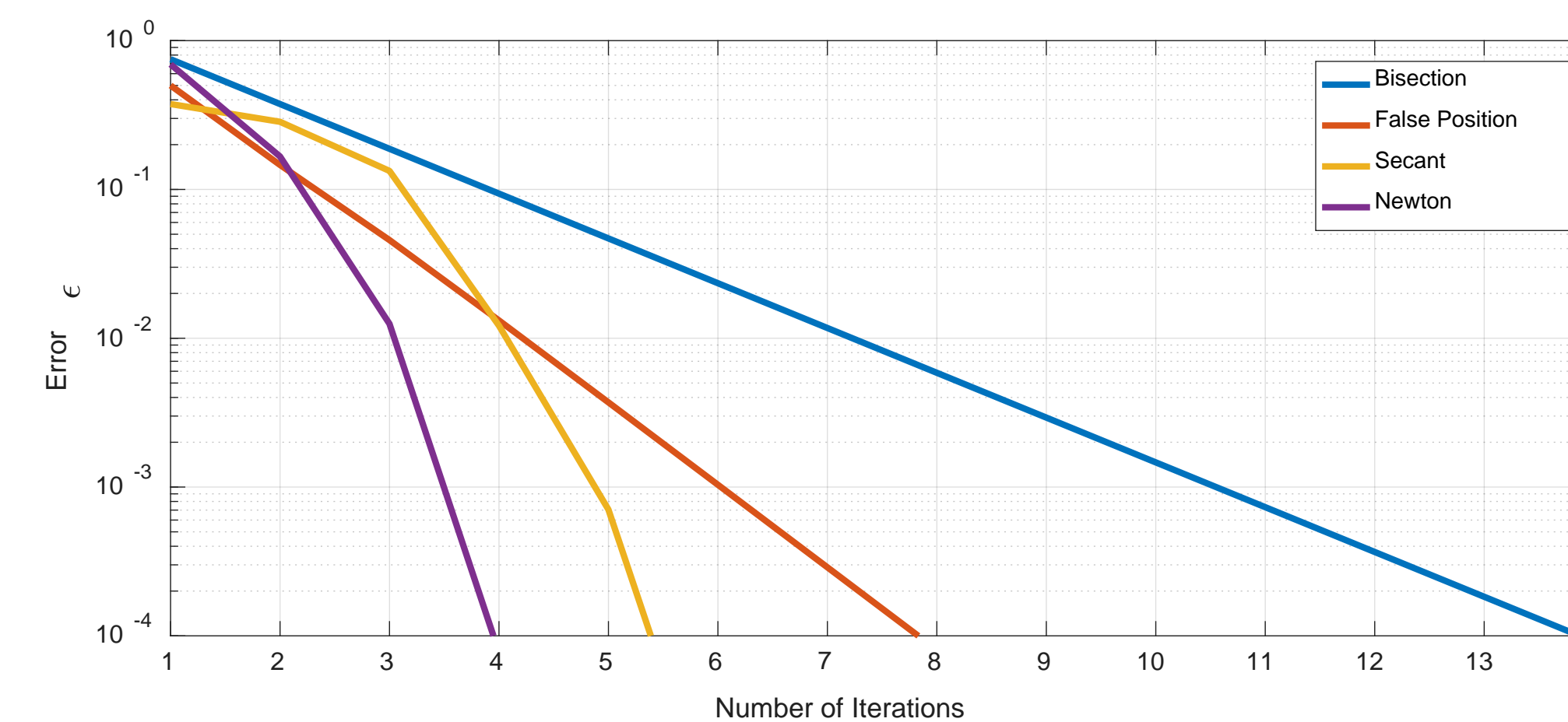


Figure 4: Plot of Relative Error per Number of Iterations for different rootfinding methods. Study performed on representative problem $0 = \ln(x)$. Illustrates the potential for optimization by using higher order rootfinding schemes. However, faster convergence is not guaranteed.

The results of the rootfinding investigation prompted the use of higher order methods for the electrochemical algorithm. Additionally the tolerancing of all iterative algorithms was modified to ensure a relative tolerance of $1e-3$ (corresponding to 99.9% accuracy), with the slice current tolerance in particular being changed to be less restrictive.

Four new configurations were developed for solving the electrochemical equations:

- False Position for equation (1)
- Secant for equation (1)
- Double Secant method where both equations (1) and (2) are modified.
- A modified Bisection Method using the relaxed tolerances was also tested.

Results

The different algorithms are compared using a test bench running MATLAB R2016b on an Intel Core i7-4770 CPU operating at 3.40 GHz. This is different from the dSpace processor used in HyPer (which also uses a compiled instead of interpretive code), however it is expected that the relative behavior illustrated in Figures (5-7) will be present.

Results

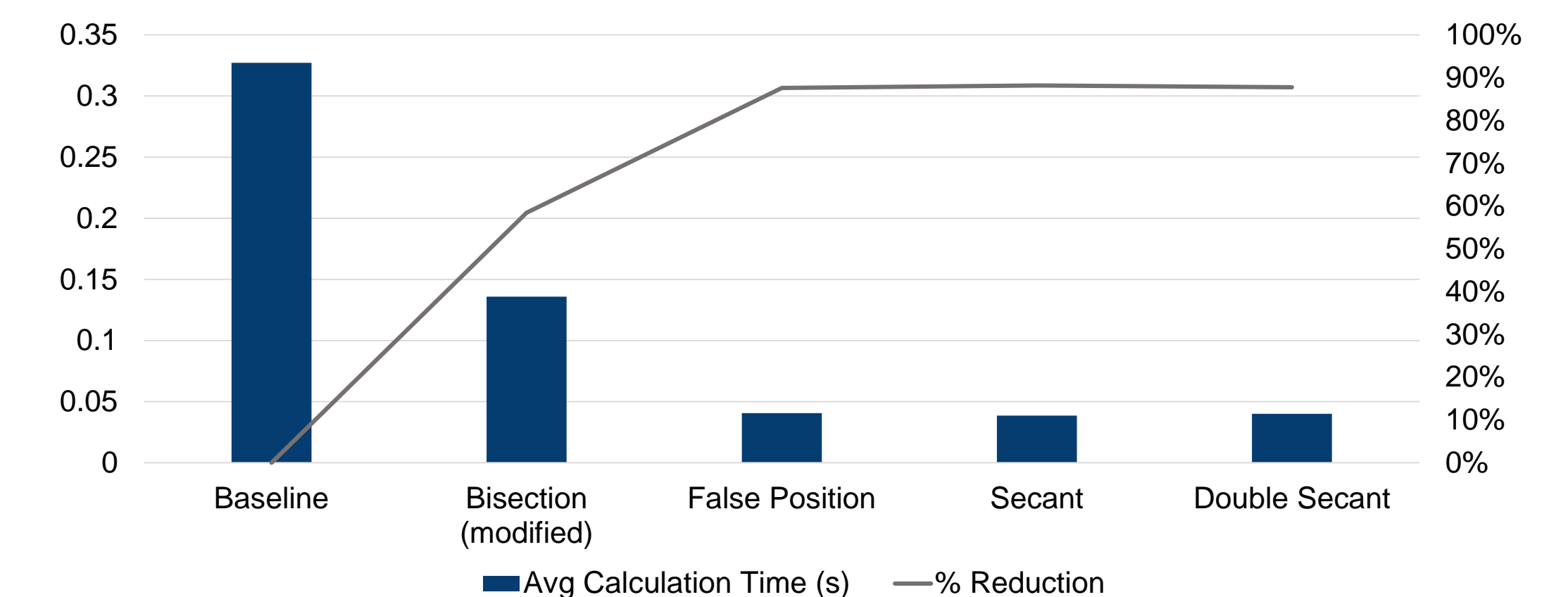


Figure 5: Plot of average calculation time in seconds for different rootfinding methods along with the percent reduction in calculation time from baseline. Model parameters are for a load of 250 A, 80% fuel utilization, time steps Δt of 40 ms, and input temperature of 1000 K.

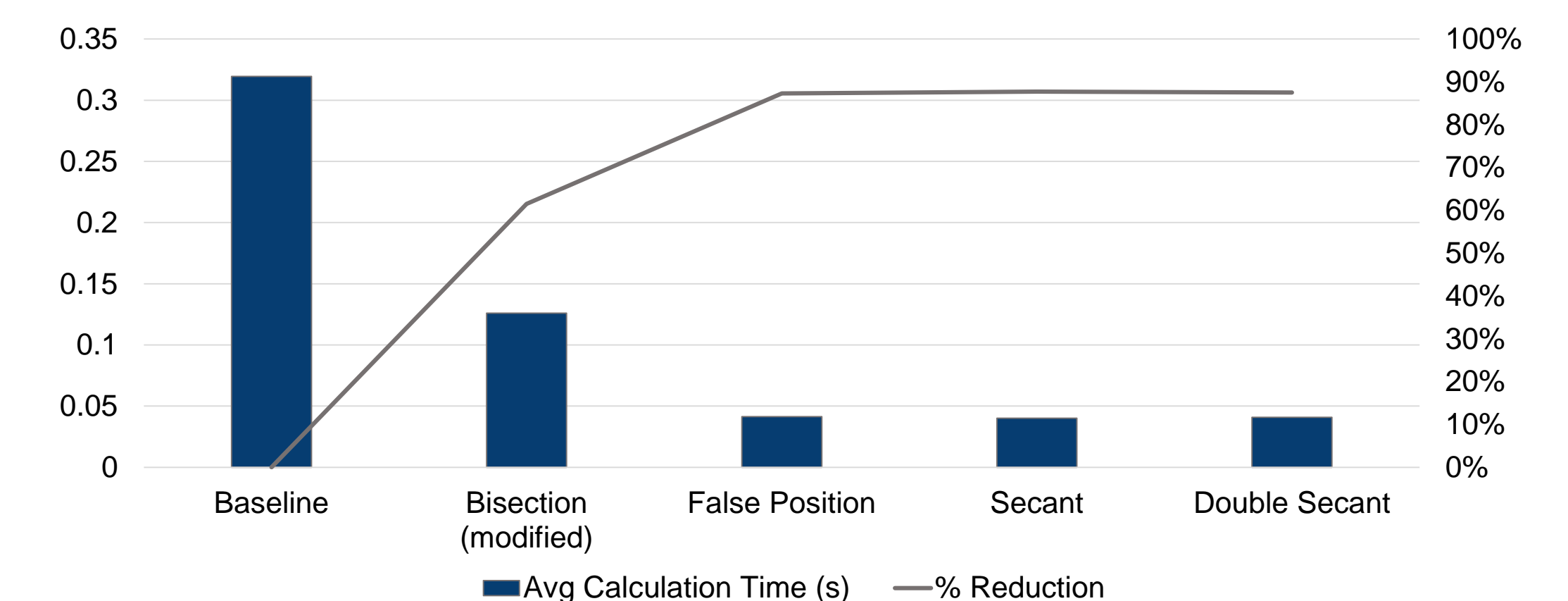


Figure 6: Plot of average calculation time in seconds for different rootfinding methods along with the percent reduction in calculation time from baseline. Model parameters are for a load of 350 A, 80% fuel utilization, time steps Δt of 40 ms, and input temperature of 1000 K.

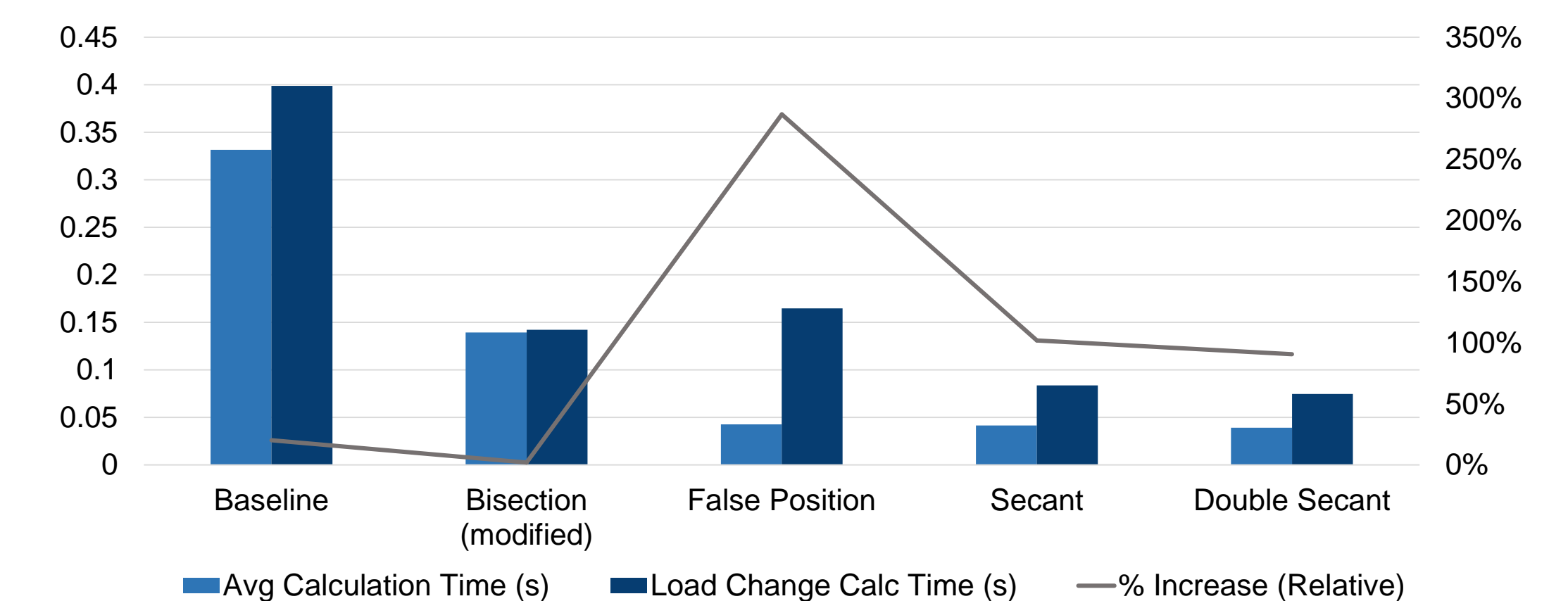


Figure 7: Plot of average calculation time and calculation time during a load step change event, both in seconds, and the percent relative increase in calculation time during the two events for each rootfinding scheme. Model parameters are for an input load of 250 A at a 50% fuel utilization that is then increased to 95% fuel utilization resulting in a final load of 450 A.

Conclusion

Overall the results indicate:

- Adjusting the tolerances significantly reduces calculation time, indicating that a bottleneck in the current density calculation was present.
- Furthermore, the higher order rootfinding schemes further accelerate code execution.
- Lastly, it can be seen that higher order rootfinding schemes are less susceptible to transient events such as step changes in current load.

Future plans include running the modified codes on HyPer to evaluate their performance on the dSpace platform.

Acknowledgements

This work is supported by the U.S. Department of Energy under Contract DE-FE-0030600

