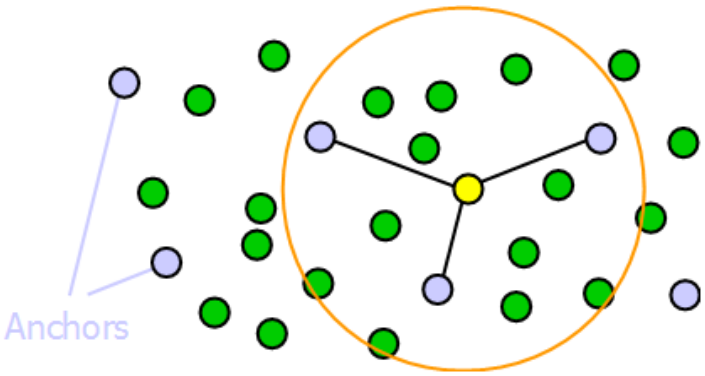
TÉCNICO LISBOA



Wireless Sensor Network Location Algorithms

Nuno Ricardo Gago Pinto

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor(s): Prof. José Carlos Campos Costa
Prof. José Carlos Alves Pereira Monteiro

Examination Committee

Chairperson: Prof. Paolo Romano
Supervisor: Prof. Carlos Nuno da Cruz Ribeiro
Member of the Committee: Prof. José Carlos Alves Pereira Monteiro

November, 2016

This thesis is dedicated to the memory of my father, António Sousa Pinto. I miss him every single day, but I am glad to know he saw this process to start, offering the support to make it possible, as well as plenty of friendly encouragement...

Acknowledgments

I would first like to thank my thesis advisor Professor José Costa of the Instituto Superior Técnico at Lisboa University. The door to Prof. Costa office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it.

I would also like to acknowledge Professor José Monteiro of the Instituto Superior Técnico at Lisboa University as the contributor to finalize this thesis, and I am gratefully indebted for his very valuable comments on this thesis.

Finally, I must express my very profound gratitude to my parents and to my partner Telma for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you!

My acknowledgment would never be complete without the special mention of my two lovely children Duarte and Alice for those important moments on your growth during the years of study. Many thanks!

Resumo

O conhecimento da localização dos sensores é um ponto fundamental nas redes de sensores sem fios. Trata-se portanto, de um dos principais requisitos para uma emorme variedade de aplicações, assim como, de aglomerações geográficos e de roteamentos. Investigações já realizadas anteriormente [1, 2] demonstam a eficácia da implementação de um método de localização para redes de sensores móveis, baseado num método amplamente reconhecido de Localização Centróide (CL) ao ar livre, inicialmente apresentado por [3] e assumindo uma localização ordenada dos sensores. No metódo de Localização Centróide [1, 2] propõem-se uma arquitectura que divide o período de amostragem original do algoritmo Centróide em janelas temporais a fim de manter o registro de informações anteriores durante o movimento, permitindo assim que existam diferentes pesos das coordenadas desses sensores. Neste trabalho, iremos investigar o método de localização ao ar livre e seus impactos sobre erro na localização, tendo em conta o consumo de energia durante o desenvolvimento de algoritmos. Com o nosso algoritmo baseado em CL, iremos apresentar as possibilidades com vista a reduzir o erro de localização e melhorar a sua precisão usando CL. Complementarmente estamos a introduzir novas abordagens de maior complexidade ao CL, considerando também uma análise do algoritmo utilizado.

Palavras-chave: Redes de sensores móveis, algoritmos de localização, localização centróide

Abstract

Localization of sensor nodes is one of the key issues in Wireless Sensor Networks. It is a precondition for a variety of applications, as well as geographic clustering and routing. Previous research on localization sensors presents [1, 2] an integrated implementation for localization algorithm for mobile wireless sensor networks based on the well-known range-free Centroid Location (CL) method, firstly presented by [3] which assumes regularly arranged beacons. The CL method [1, 2] proposes an architecture that splits the original sampling period of the Centroid algorithm into temporal windows in order to maintain a record of past information during movement, allowing for the weighting of the anchors' coordinates. In this work, we investigate the range-free location method and its impacts on localization error, keeping in mind the power consumption during the algorithm development. With our based CL algorithms, we present possibilities to reduce the location error and improve their accuracy using CL. However, since our new approaches introduce more complexity to CL, an analysis of the algorithms usage is considered.

Keywords: mobile sensor network, range-free localization algorithm, centroid location

Contents

List of Tables

List of Figures

Nomenclature

AOA	Angle of Arrival
API	Application Programming Interface
CL	Centroid Location
CPU	Central Processing Unit
GPS	Global Positioning System
IoT	Internet of Things
MAC	Medium Access Control
OS	Operating System
RF	Radio Frequency
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indicator
TDOA	Time Difference of Arrival
WLAN	wireless local area network
WSN	Wireless Sensor Networks

Chapter 1

Introduction

Today, sensors are everywhere. Smart grid, smart homes, smart water networks, intelligent transportation, are infrastructure systems connected to our world more than we ever thought possible. The common vision of such systems is usually associated with one single concept, the internet of things (IoT), where through the use of sensors, the entire physical infrastructure is closely coupled with information and communication technologies; where intelligent monitoring and management can be achieved via the usage of networked embedded devices. In such a sophisticated dynamic system, devices are interconnected to transmit useful measurement information and control instructions via distributed sensor networks.

A wireless sensor network (WSN) is a network formed by a large number of sensor nodes where each node is equipped with a sensor to detect physical phenomena such as light, heat, pressure, etc. A typical sensor node consists of four basic components: sensing, processing, transceiver (transmitter and receiver) and power units (usually, a battery), as illustrated in Figure 1.1.

Technological advances led to the development of tiny wireless devices, which are able to sense the environment, compute simple tasks and exchange data among each other. Combining the advantages of wireless communication with some computational capabilities, WSNs allow for a wider variety

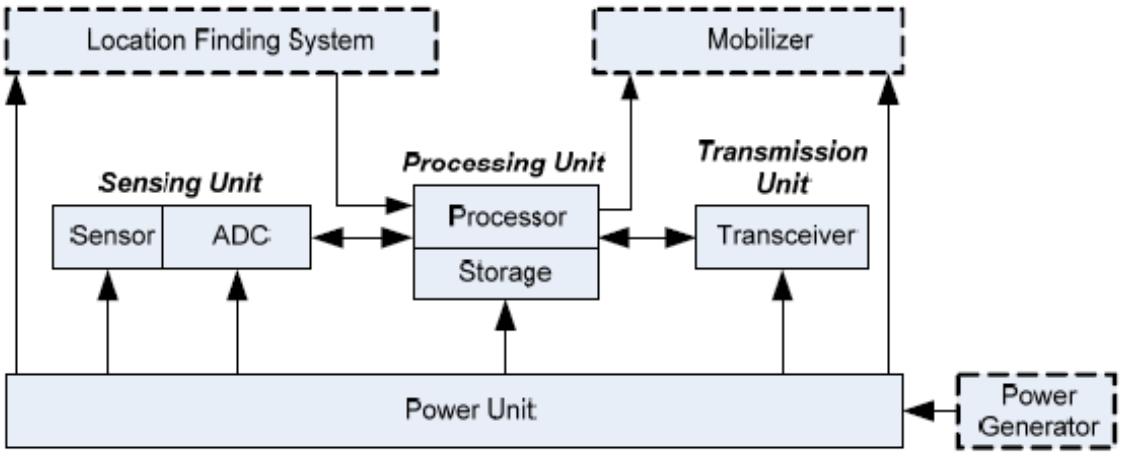


Figure 1.1: Typical wireless sensor node with their components.

of applications than traditional networks: environmental monitoring, health, surveillance, catastrophe monitoring, structural monitoring, security, military, industry, agriculture, home, traffic monitoring, and others.

1.1 Localization in Wireless Sensors Networks

In WSNs the location estimation may enable a myriad of applications such as inventory management, transport, intrusion detection, road traffic monitoring, health monitoring, reconnaissance and surveillance.

One of the most know and common methods is the commercial positioning system GPS (Global Positioning System) [4] as broadly used in our cars, although GPS based localization estimation has well known limitations in terms of size and high energy consumption, making localization within the sensors network the preferred method over the GPS. Moreover, it cannot be used indoors, because of insufficient GPS coverage.

Therefore, assuming the localization within the wireless sensor network the preferred method and, since, the wireless links between the sensors have highly irregular and probabilistic properties. This way, link quality can vary significantly over time, especially in indoor environments due to human activity and multi-path or reflexion effects. Consequently, a typical approach is to assume that only a small number of selected nodes know their exact position a-priori or obtain the position using common positioning systems. Then, all other nodes calculate their position with the help of these beacon nodes.

Localization within the wireless sensor network algorithms estimate the locations of sensors with initially unknown location information by using knowledge of the relative positions for a few sensors and inter-sensor measurements such as distance and bearing measurements. Location techniques based on radio signal propagation and sensors communication in WSN can be split in three categories: Angle-of-arrival (AOA) measurements, distance related measurements, and Received Signal Strength (RSS) profiling techniques, explained in more detail later in Chapter 2.1.

The accuracy of localization techniques ranges from high precision, commonly based on solving a set of nonlinear equations, to low precision. Localization methods can be divided into coarse-grained and fine-grained algorithms, here we propose a coarse-grained localization algorithm, which needs only a minimum of computations, called Centroid Localization. In the Centroid Localization method, all non-localized nodes calculate their own position based on the centroid of all received positions from the other beacons, where the most reliable ones are the Base Stations, where their position is fixed and known in advance.

The pure Centroid Location (CL) [1, 3, 5] method does not utilize the Received Signal Strength Indicator (RSSI) neither Angle-of-arrival (AOA) measurements, neither any other parameter indicating the distance between a beacon node and an unknown one. The only consideration used by the Centroid Location method to calculate the distance information in Centroid Location is the binary information whether the unknown node is in the communication range of a beacon or not, where the location in known of this beacon in advance. Centroid Location assumes a circular area with the center being the

beacon's location as communication range.

The CL method assumes a circular area with the center being the beacon's location as communication range, i.e. unit disc graph model. The CL uses the location information of all beacons in range, of the sensor with unknown location, to calculate the position as the centroid of the received beacon positions. This approach has some pitfalls that can occur as the following extremes: the beacons listens to all the possible neighbors, since their radio signal covers all the map, in this case the estimated position for all sensors is in the center of the map; the beacons cannot listen to any neighbor or beacon, meaning the sensors are completely without any coverage, in this case the sensors cannot calculate their position.

To overcome these extreme case scenarios and even estimate the location of sensors which are out of a coverage from any beacon, a proposed approach, a called Modified Centroid Location, introducing historical information on the calculated position of the beacons is introduced and verified using different densities on the number Base Stations nodes (the nodes which are on fixed and known positions), and different coverage areas. It has also been verified the impact of having different weights (linear and exponential) on the historical information for the study performed.

1.2 Motivation

With the advent of short range wireless technologies and standards in late 1990's a variety of wireless localization techniques for indoor and outdoor applications have been developed. Wide range of indoor localization techniques have emerged based on camera, infrared, wireless local area network (WLAN), ultra wide band (UWB), Bluetooth, and radio-frequency identification (RFID) [6] whereas global positioning system (GPS) technology revolutionized outdoor localization.

Localization techniques have been developed for different types of applications and are compared in terms of accuracy, coverage, cost, responsiveness and adaptiveness to environmental changes.

The algorithm proposed here using Centroid Location methodology can be designed for applications such as pest control, irrigation in large farms, orchards where greater power efficiency and scalability are required but location accuracy requirements are less demanding. Our algorithm uses received position from the nodes within their coverage are and estimates their position using the Centroid Method including the historical positions information.

The previous researches that conducted to the Centroid Location methodology [1, 3, 5], used as basis for the current dissertation, do not consider the challenges caused by the movement of the sensors, on a free range mobility, for localization estimation. This dissertation evaluates the solutions to enhance accuracy of the Centroid Location method when the sensors move on a free range environment.

The CL requires some sensor nodes to be anchor nodes, which are nodes with known locations, deployed uniformly throughout the network such that non-anchor nodes should always have radio connectivity to several anchors at once. The locations of the non-anchor nodes can then be computed by the Centroid algorithm [1, 3, 5] which localizes a node to the average of the coordinates of all anchors with which it has connectivity.

The range-free localization approach considered in this dissertation is based on the observation that

nodes with radio connectivity are typically in close proximity while those without radio connectivity are not. Because it only relies on the use of a radio, similar approach can be applied to any other wireless device.

1.3 Objectives to Evaluate the Modified Centroid

The centroid algorithm can be used to estimate coordinates from the coordinates of the neighbor nodes within the radio signal range. The variant studied by this dissertation considers the nodes are moving within a known area where some nodes are fixed with known positions considered by this as Base Stations providing the coordinates for the rest of the cluster network, thus giving an indication of the direction of movement.

For this dissertation the Modified Centroid algorithm was implemented in TinyOS and sensors are simulated in TOSSIM networks (TinyOS mote SIMULATOR). TinyOS is an operating system designed for event-driven sensor networks and have very limited resources. TinyOS is ”open source” and is designed for wireless devices with low power consumption, such as sensor networks, ubiquitous computing and intelligent buildings.

1.4 Thesis Outline

In this dissertation, we review the state of art for node localization in Wireless Sensor Networks and show how the localization protocol has the potential to provide the robust solution that is needed. The research challenge that we face is how to obtain a highly accurate node locations in large scale sensor networks deployed in complex environments, at the lowest cost possible.

In this dissertation, we present our investigations concerning the CL and the achievable improvements done on the location accuracy, when the RSSI method is introduced by weighting the location retrieved from the Base Stations. Our approach will be evaluated in terms of accuracy and power consumption of the motes. Since our major focus is on the Centroid Methodologies it was evaluated the impact when introducing historical information to this method.

The remainder of this report is organized as follows:

In the next chapter, we emphasize the main objectives we would like to archive an the methods used as base for our approach as the references analyzed and state of art for node localization in Wireless Sensor Networks algorithms.

In chapter 3 is presented our proposed modifications to the Centroid algorithm for both static and mobile networks and described the architecture used to develop and perform our experiments to verify our Centroid Location approach.

In chapter 4 we explain the experiments performed and their results in detail.

Chapter 5 presents the conclusions retrieved from the experiments performed and the we discuss the future developments to improve the results.

Chapter 2

Related Work

The work that has been done on sensor localization algorithms can be classified in two main categories: the range-based approaches and the range-free approaches. These approaches differ from each other essentially in the way distance information is obtained. Range-based approaches are based in distance or angle measurements, requiring the installation of specific and expensive hardware (e.g., directive antennas). Range-free approaches only consider connectivity information between adjacent nodes [7, 8], and also referenced in the Centroid Location [1, 2].

However, there are also hybrid solutions, which combine the advantages of range-based approaches with the advantages of range-free approaches. Moreover, there is either range-based or range-free approaches combined with the use of anchor nodes, based in MDS (Multidimensional Scaling), centralized or distributed, or mobile-assisted [7] and proximity based map (PDM) [8] or MDS and Ad-hoc Positioning System (APS) [9]. These techniques contribute to new directions in WSN localization as these schemes give high accuracy in low communication and computation cost. On the other hand interferometric ranging based localization has been proposed also [10, 11, 12].

Furthermore, due to channel fading and noise corruption error propagation comes in the picture. To suppress this error propagation a localization scheme has been proposed in [13] which has not been discussed by any literature. This literature gives comprehensive summary of these techniques along with other existing localization schemes.

2.1 State of the Art

Existing localization discovery approaches generally consists of two basic phases: (1) distance (or angle) estimation and (2) distance (or angle) combining, towards the beacon. The most used methods for estimating the distance between two nodes are described below.

Received Signal Strength Indicator (RSSI): RSSI measures the power of the radio signal at the receiver and based on the known transmit power, is possible to determine the effective propagation loss. Next by using theoretical and empirical models this loss can be translated into a distance estimate. RSSI is a relatively cheap solution without any extra hardware, as all the radio of sensor nodes are

likely to have this functionality included. The distances measured with RSSI have error due to the multipath propagation of radio signals. In [14], the authors describe in detail a variety of approaches to indoor localization using signal strengths from 802.11 routers. It is also suggested that adding additional hardware or altering the model of the environment is the only alternative to improve the localization performance.

Time based methods (ToA, TDoA): These methods record the time-of-arrival (ToA) or time-difference-of-arrival (TDoA). The propagation time can be easily converted into distance, based on the known signal propagation speed. These methodologies can be applied to many different signals, such as RF, acoustic, infrared and ultrasound. TDoA methods are impressively accurate under line-of-sight conditions. But this line-of-sight condition is difficult to meet in majority of environments. Furthermore, the speed of sound in air varies with air temperature and humidity, which introduce error into distance estimation. Acoustic signals also show multi-path propagation effects that impact the signal detection accuracy.

Angle-of-Arrival (AoA): AoA estimates the angle at which signals are received and use simple geometric relationships to calculate node positions. Generally, AoA techniques provide more accurate localization result than RSSI based techniques but the cost of hardware of very high.

Centralized Localization: Centralized localization is basically migration of inter-node ranging and connectivity data to a sufficiently powerful central base station and then the migration of resulting locations back to respective nodes. Therefore, the data collected by all nodes in the network is sent to a central processing server. So, the central processor collects all inter-node distance measurements, being responsible to compute the localization map of the entire network. The advantage of centralized algorithms is that it eliminates the problem of computation in each single node, although at the same time, the limitation lie in the communication cost of moving data back to the server unit.

Distributed Localization: In Distributed localizations all the relevant computations are performed on the sensor nodes themselves and the nodes communicate with each other to get their positions in a network, therefore the localization task relies on the nodes themselves (self-localization). Localization is performed based in inter-node distance measurements and in information collected from close neighbors.

Furthermore, [1, 2] proposed a method based on the Centroid algorithm that targets mobile networks, where the original sampling period of the Centroid algorithm is divided into temporal windows in order to maintain a record of past information of the node during movement. The selection of the anchor nodes is based on the recorded history, allowing for the weighing of the anchor's coordinates in the localization process. For example, if the number of messages received from an anchor is decreasing that is an indication that the node is moving away from it. Even if the target were mobile networks, with or without control movement, the method also proved to increase the accuracy of the Centroid algorithm in static networks.

2.2 Localization Method

Extensive work has also been done on modeling RF propagation and deploying systems based on received signal strengths [15, 16, 17, 18]. These systems require mapping signal strengths to distances based on known radio characteristics. This type of approach is highly dependent on signal propagation in the environment as well as individual antenna and receiver specific properties.

The research reported in [3] describes a range-free localization technique where each mobile node derives the position by calculating the center of the location of all beacons it hears. If beacons are placed well, localization errors can be decreased [19], but this is not possible in all environments. One option to improve the pure centroid algorithm would be to add an RSSI factor to the location computed.

Utilizing information from the environment would perhaps be the most basic and frequently overlooked source of localization information. Many of these environmental features such as background noises slowly changes over time, hence it becomes significantly difficult to uniquely calculate a large number of locations. Therefore, it requires the use instantaneous time synchronized sensor features to correlate with nearby locations. Hence, we do not require a database and the approach does not suffer from slow changes in the environment over time, and this way reduces the error.

The accuracy of localization techniques ranges from high precision, commonly based on solving a set of nonlinear equations, to low precision. [3] divided localization into coarse-grained and fine-grained algorithms and proposed a coarse grained localization algorithm, which needs only a minimum of computations, called Centroid Localization (CL) [4]. In CL, all non-localized nodes calculate their position as the centroid of all received beacon's positions. In [19] studied the precision of CL.

2.3 Centroid Localization

The pure CL does not utilize the Received Signal Strength Indicator (RSSI) or any other parameter, indicating the distance between a beacon node and an unknown. The only kind of distance information used in CL is the binary information whether the unknown node is in the communication range of a beacon or not. CL assumes a circular area with the center being the beacon's location as communication range, i.e. unit disc graph model. Figure 2.1 depicts the communication ranges of four beacons, arranged as described above. It is shown that thirteen Intersection Areas (IAs) can be distinguished where an unknown node can be localized.

The CL uses the location information of all beacons in range to calculate the position as the centroid of the received beacon positions, as shown in equation 2.1. Here, $P_i(x, y)$ indicates the position of unknown node i given by the two dimensional coordinates. The known position of beacon j is given by $B_j(x, y)$. The number of beacons which are within the communication range of the unknown node is indicated by m.

$$P_i\left(x, y\right) = \frac{1}{m} \sum_{j=1}^{m} B_j\left(x, y\right) \tag{2.1}$$

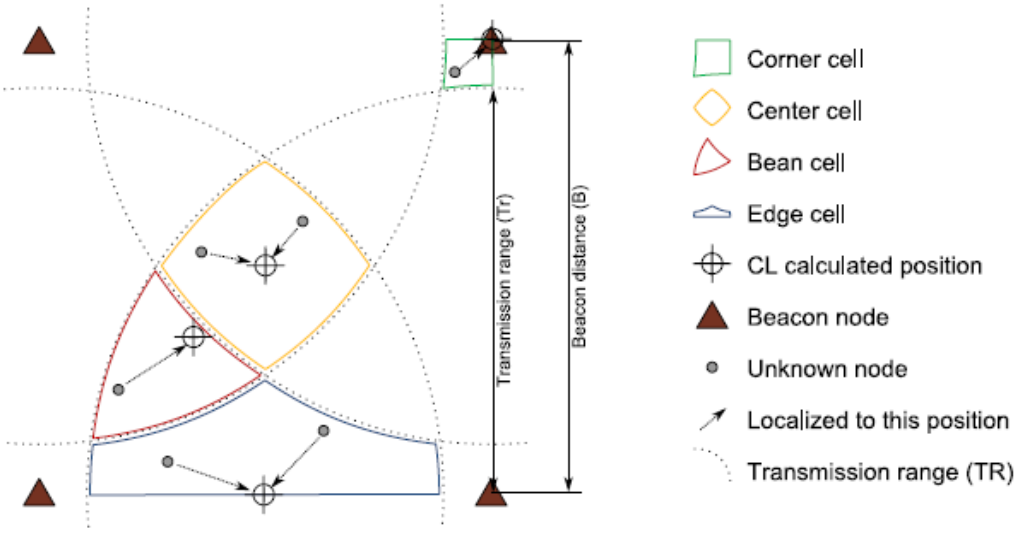


Figure 2.1: On Centroid Location nodes assign themselves to a cell by listening to messages from the beacons. Then, the node localizes itself at the estimated centroid of the according cell.

A node which is situated within one of the interception areas will calculate the position at one single point, regardless of their exact position within the interception area. For each interception areas exists such an localization point which is the centroid of the beacon positions in range. This behavior leads to a relatively high localization error, given as the Euclidian distance between the exact position of a sensor node and the calculated position. This is the main concept of the model proposed by us.

2.4 Received Signal Strength Indicator

To overcome the location error, the Received Signal Strength Indicator (RSSI) could be considered, and it would help on the accuracy for the position. In [20], the authors propose a scheme which localizes nodes through RF attenuation in electromagnetic waves. The scheme proposed basically consists of three stages:

- RF mapping of the network: It is obtained by conveying short packets at different power levels through the network and by storing the average RSSI value of the received packets in memory tables.
- Creation of the ranging model: All the tuples recorded between the two anchors are processed at the central unit to compensate the non linearity and calibrate the model. Let a generic tuple (i, j, P_{tx}, P_{rx}) comes from the RF mapping characterizing stage, where i is the transmitting node and j is the receiving node. Now first the algorithm corrects the received power as

$$P'_{rx} = f\left(P_{rx}, P_{tx}\right) \tag{2.2}$$

where $f()$ is a function which takes into account the modularity effects. So, the estimated distance between the nodes will be

$$r_{i,j}^{0} = m^{-1}\left(P'_{rx}\right) \tag{2.3}$$

- Centralized localization model: An optimization problem is solved and provides the position of the nodes. The final result can be obtained by minimizing the function

$$\begin{cases} E = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(k_{i,j}a_{i,j}\left(r_{i,j} - r_{i,j}^{0}\right)^{2}\right) \\ r_{i,j} = d\left(i,j\right) \end{cases} \text{, when } i,j \text{ are anchores} \tag{2.4}$$

  Where N is the number of nodes, $a_{i,j}$ is 1 when the link is present and 0 otherwise.

  Once the distance between the nodes r_{ij} can be expressed in terms of their coordinates $(x,y)_i$ and $(x,y)_j$ the authors solve the minimizing problem by Sequential Quadratic Programming (SQP) method. The advantage of this scheme is that it is a practical, self-organizing scheme that allows addressing any outdoor environments. The limitation of this scheme is that the scheme is power consuming because it requires extensive generation and need to forward much information to the central unit. To overcome these issues (power consumption and the centralized point) it can be proposed to perform a mix of both methods, the pure Centroid Location being weighted with the RSSI.

2.5 Radio Medium Access Control

On our simple experiments it has been seen communication issues caused by the ad-hoc radio usage on the motes, which also would increase the error on the motes communication. To overcome the radio interference, a Medium Access Control procedure should be implemented to guarantee the uniqueness radio access for a single node.

Several Medium Access Control (MAC) protocols have been considered, as references bases for this development, keeping in mind the low-power and distributed operation for single and multi-hop wireless mesh networks. Such protocols may be categorized by their usage of time synchronization as asynchronous, loosely synchronous and fully synchronized protocols. In general, with a greater degree of synchronization between nodes, packet delivery is more energy-efficient due to the minimization of idle listening when there is no communication, better collision avoidance and elimination of overhearing of neighbor communications.

The Berkeley MAC (B-MAC) [21] protocol is an energy efficient and simple to implement extension upon standard carrier sense multiple access (CSMA) wireless communication, similar to the one broghtly used at the WIFI routers. BMAC supports CSMA with low power listening (LPL) where each node periodically wakes up after a sample interval and checks the channel for activity for a short duration of miliseconds. If the channel is found to be active, the node stays awake to receive the payload following an extended preamble. Using this scheme, nodes may efficiently check for neighbor activity. For each transmission instance, the transmitter must remain active for the duration of the receiver's channel check

interval. This creates a trade-off between listening more frequently or transmitting longer. For a particular sampling rate, one can calculate the optimal radio check rate.

Protocols such as S-MAC [22] and T-MAC [23] employ local sleep-wake schedules know as virtual clustering between node pairs to coordinate packet exchanges while reducing idle operation. Both schemes exchange synchronizing packets to inform their neighbors of the interval until their next activity and use CSMA prior to transmissions. As all the neighbors of a node cannot hear each other, each node must set multiple wake-up schedules for different groups of neighbors. The use of CSMA and loose synchronization trades energy consumption for simplicity.

In our case, the Base Stations (the fixed nodes with known location) send broadcast messages with their position in known time schedules. The broadcast allows the dissemination transmission with the maximum power setting among all the node's neighbors. This policy ensures that all neighbors can receive the message. This Broadcast message, will trigger the Mobile Nodes, to exchange some messages with the Base Stations to extrapolate a RF model and reduce the error.

Several methodologies have been referenced to model RF propagation and deployment based on received signal strengths, these systems require mapping signal strengths to distances based on known radio characteristics [24, 22, 21], these type of approaches are highly dependent on signal propagation in the environment as well as individual antenna and receiver specific properties. We use the site survey database approach to establish a baseline to which we compare our reference-free and micro-climate-based approaches and this way improve the centroid algorithm. Hence, we do not require a database and the approach does not suffer from slow changes in the environment over time, and simply is composed of the average received signal strength indication (RSSI) values received by beacons.

The weighted-centroid approach does not require an RF propagation model or site survey it only requires the placement of some beacons in the environment at known coordinates. RSSI values received from each node are sorted and then the highest RSSI values are used to find the centroid of the convex polygon that falls within the selected beacon node locations. The notation is referred as location of a mobile node is denoted by $Mobile_{(x,y)}$, and comprises of its (x, y) coordinates represented by $Mobile_x$ and $Mobile_y$ respectively. $Beacon[i]$ is the ith element of the list of beacons that are in range of the mobile node. $Beacon[i]_{rssi}$ is the RSSI value between the mobile node and beacon i. $w[i]$ is a weight factor that can be associated with each RSSI value. It may also be choose to apply different weight factors $w[i]_x$ and $w[i]_y$ for the x and y coordinate dimensions respectively. When we adopt the RF-only localization technique, $w[i]$ has a value of 1, but it will be assigned other values in later sections.

$$Mobile_{(x,y)} = (Mobile_x, Mobile_y) \tag{2.5}$$

$$Beacon_{(x,y)} = (Beacon_x, Beacon_y) \tag{2.6}$$

$$RSSI_{total} = \sum_{i=0}^{m} (Beacon\,[i]_{rssi} \times w\,[i]) \tag{2.7}$$

$$Mobile_x = \sum_{i=0}^{m} \left(\frac{Beacon\,[i]_{rssi} \times w\,[i]}{RSSI_{total}} \times w\,[i]_x \times Beacon\,[i]_x \right) \tag{2.8}$$

Since the centroid will always be bounded within the beacon nodes, mobile nodes cannot be accurately tracked once they travel outside the perimeter of the beacons. This seems to be a reasonable trade-off given the computational efficiency as compared to other trilateration techniques. The weighted-centroid approach does not require RSSI values to be converted into a distance. In an environment where the path-loss component can vary dramatically, this method helps mitigate errors by weighting distance relative to all nearby RSSI values.

Chapter 3

Centroid Modified Location

The Centroid Location (CL) Method assumes a circular area with the center being the beacon position and the radius the circular area defined by the communication range, i.e. unit disc graph model. The CL uses the localization information from all beacons in the coverage range to compute the position as the centroid of the received beacons positions. In Figure 2.1 it can be visualized the CL method approach and it is shown the Intersection Areas (IAs) where an unknown node can be localized.

The CL method may have the following limit cases: the sensors can listens to all beacons existent on map, because the beacons radio signal have enough power that covers all the area in the map, in which the motes calculated position (the center of the IA) is in the center of the map or the center of the Base Stations covered area; the motes can not listen to any neighbor, hence the sensors cannot calculate their position.

Moreover, the CL methodology can only be used when the sensors are static, if the sensors are moving the CL method needs to have samples of the position. The method proposed in the current dissertation approaches the introduction of historical information for the computation of the position for the motes to overcome the limitations of the pure Centroid Location, referred here as Modified Centroid Location. To verify the approach different densities on the number Base Stations (beacons) nodes (the nodes which are on fixed and known positions broadcasting their location) within the map, and also using different coverage areas (signal power), were included. Furthermore, it has also been verified the impact of having different historical weights values (uniform, linear and exponential) on the study performed.

The test environment used in all the cases was created on a limited area where the nodes perform the movement and calculates the position relative to the beacons with known position which are considered the Base Stations. This limited map area has dimensions 650×650. Also, in all the simulations has been considered 9 nodes moving and on several densities for the Base Stations (number of beacons): 8, 16, 25, 36, 49, 56, 64, 72, 81, 100; using different coverage areas: 100, 200, 300, 400.

Furthermore, it has been verified the impact of the historical information on the error of the calculated location and also impacts when implementing the intercommunication between motes making by this way all the nodes beacons.

The simulation environment has been implemented using TinyOS [25], a free and open source soft-

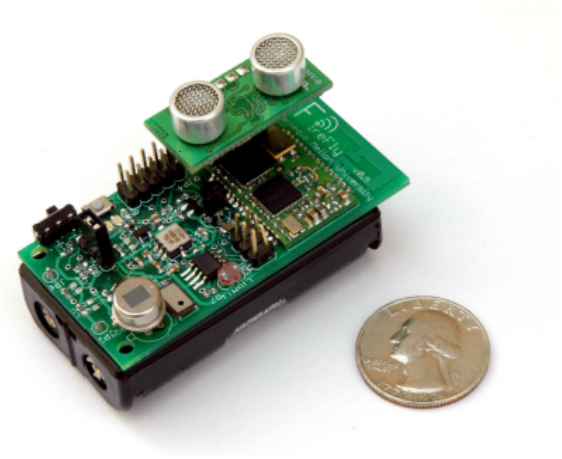


Figure 3.1: Sensor node example.

ware component-based operating system and platform targeting wireless sensor networks, with low footprint “operating system” that supports modularity and concurrency using an event-driven approach.

3.1 Test Environment

All experiments were performed on a testbed consisting of TelosB motes equipped with CC2420 radios using TinyOS 2.x’s default CSMA/CA MAC layer. The CC2420 radio chip can be programmed to operate at 8 different power levels with output power ranging from -25 dBm to 0 dBm and current consumption ranging from 8.5 mA to 17.4 mA [24]. The CC2420 radio also provides the possibility for an RSSI reading and LQI reading embedded in the metadata of all the incoming packets. The RSSI reading represents a sampling of the signal strength (transmission + background noise) taken at the beginning of the packet reception, while the LQI reading represents the average symbol correlation value over the packets exchange per inter-connection node.

For our implementation each Base Station node broadcasts packets while the neighbors sensors with unknown localization record the Originator ID, packet sequence number, localization position, RSSI reading, and LQI reading of all packets received. According to the TinyOS description the proposed environment used on our approach is a TDMA-based link layer protocol designed for networks that require predictability in throughput, latency and energy consumption, like our case.

After research to review the state of art for node localization in Wireless Sensor Networks, as referenced in Chapter 2.1, a simple network example, using TOSIM (TinyOS mote SIMulator) based on TinyOS 2.x of TelosB motes equipped with CC2420 radios, has been used to implement the different levels of evolution for our experiments.

TOSSIM is a discrete event simulator for TinyOS sensor networks. Instead of compiling a TinyOS application for a bare metal mote, the TOSSIM framework can be compiled on a normal PC. This implementation allows debugging, testing, and extend analyzes of the algorithms implemented in a controlled and repeatable environment. Figure 3.2 shows the graphical overview of TOSSIM architecture.

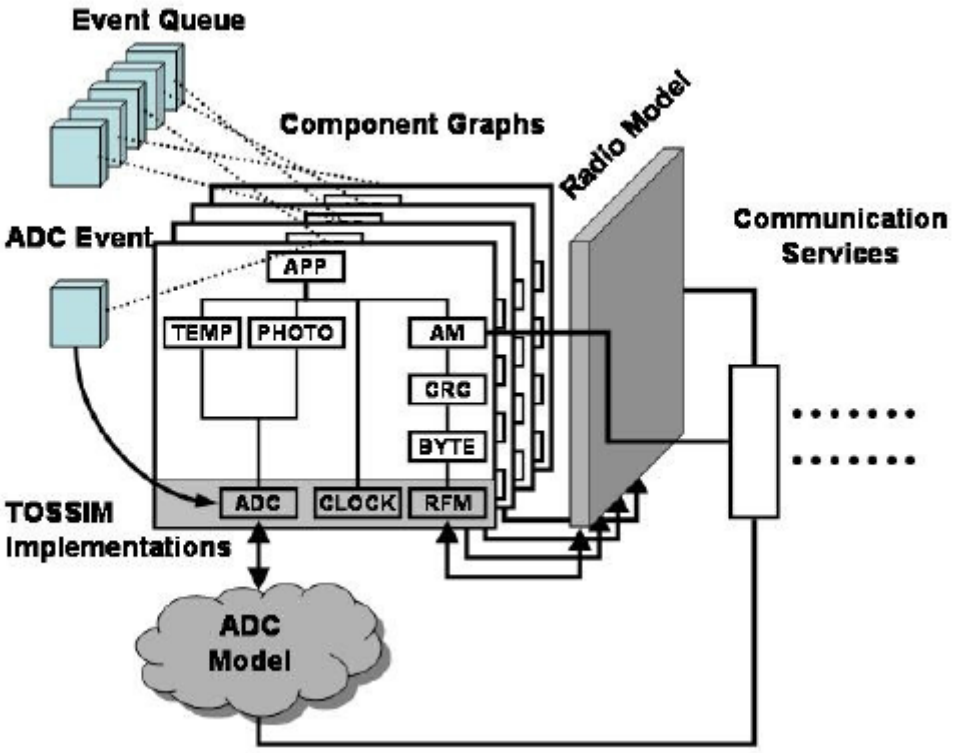


Figure 3.2: TOSSIM Architecture components.

The main reasons for TOSSIM choice on our test environment are based on the following main characteristics:

Fidelity: By default, TOSSIM captures TinyOS behavior at a very low level. It simulates the network at the bit level, simulates each individual Analog-to-Digital Converters capture, and every interrupt in the system. Allowing a repeatable and a full understanding of the computational information to experiment our model.

Time: While TOSSIM precisely times interrupts (allowing things like bit-level radio simulation), it does not model execution time. From TOSSIM's perspective, a piece of code runs instantaneously. Time is kept at a 4MHz granularity (the CPU clock rate of the rene and mica platforms). This also means that spin locks or task spin locks will never exit: as the code runs instantaneously, the event that would allow the spin to stop will not occur until the code completes (never), allowing to simulate the motes movement.

Models: TOSSIM itself does not model the real world. Instead, provides abstractions of certain real-world phenomena (such as bit error). With tools outside the simulation itself, can then be manipulated these abstractions to implement whatever models are needed to be used. By making complex models exterior to the simulation, TOSSIM remains flexible to the needs. Additionally, it keeps the simulation simple and efficient.

A TinyOS program is composed of components, which are independent computational entities. Components have three computational concepts: commands, events, and tasks. Commands and events are used for inter-component communication, while tasks provide intra-component concurrency. The component model, which is designed in TinyOS, allows the target platform to be easily changed from mote

hardware to simulation mode, which could be a next step to verify out model on a real environment scenario without the need to change any line of code. TOSSIM models the characteristics of the underlying hardware in software. To model hardware interrupts, a simulator event queue is located in TOSSIM that delivers the interrupts.

3.2 Operating System used - TinyOS

TinyOS is a lightweight operating system explicitly designed for low-power wireless sensors. TinyOS differs from most other operating systems in that its design focuses on ultra low-power operation. Rather than a full-fledged processor, TinyOS is designed for the small, low-power microcontrollers motes. Moreover, TinyOS has features and mechanisms very focused on saving power.

TinyOS allows building sensor network applications easier. It has built-in a set of important services and abstractions, such as sensing, communication, storage, and timers. It defines a concurrent execution model, so developers can build applications out of reusable services and components without having to worry about unforeseen interactions. TinyOS runs on generic platforms, most of which easily support adding new sensors. Furthermore, TinyOS's structure makes reasonably easy to port to new platforms. TinyOS applications and systems, as well as the OS itself, are written in the nesC language [26]. nesC is a C dialect with features to reduce RAM and code size, enabling significant optimizations, and helping to prevent low-level bugs like race conditions. The component model is grounded in nesC. It allows written pieces of reusable code which explicitly declare their dependencies. For example, a generic user button component that tells when a button is pressed sits on top of an interrupt handler. The component model allows the button implementation to be independent of which interrupt that is - e.g., so it can be used on many different hardware platforms - without requiring complex callbacks or extra function naming conventions.

The concurrent execution model enables TinyOS to support many components needing to act at the same time while demanding little RAM. First, every I/O call in TinyOS is split-phase: rather than block until completion, a request returns immediately and the caller gets a callback when the I/O completes. Since the stack is not tied up waiting for I/O calls to complete, TinyOS only needs one stack, and does not have threads; Instead, uses tasks, which are lightweight deferred procedure calls. Any component can post a task on the kind-of queue, which TinyOS will run at some later time. Because low-power devices must spend most of their time asleep, therefore having low CPU utilization and so in practice tasks tend to run very soon after they are posted (within a few milliseconds). Furthermore, because tasks can not preempt each other, task code does not need to worry about data races. Finally, TinyOS itself has a set of APIs for common functionality, such as sending packets, reading sensors, and responding to events, which are broadly used on our model.

Once, TinyOS supports a cyclic-executive model wherein interrupts can register events, which can then be acted upon other non-blocking functions. The infrastructure elements implemented with TinyOS are capable of providing QoS properties to a wide variety of sensor networking applications. The system's functionality is evaluated as part of an extensible sensing network, and this way help the Centroid

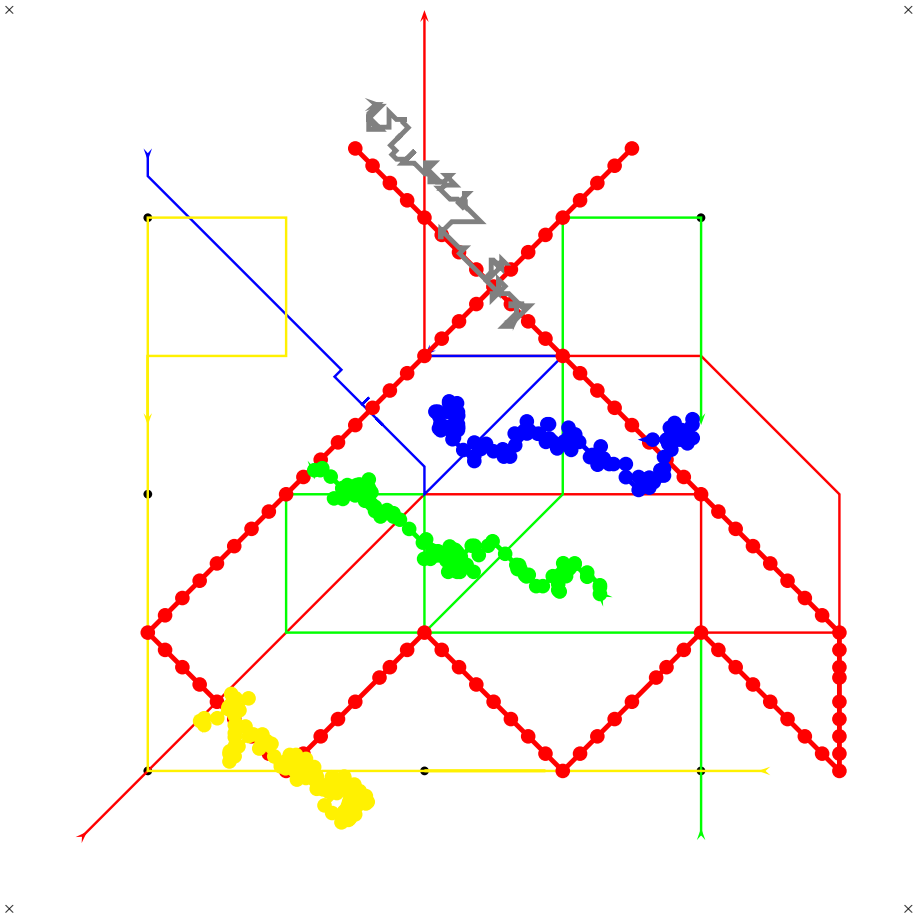


Figure 3.3: Movement of the motes on the simulation environment: Mote 1 in red (thin); Mote 2 in green (thin); Mote 3 in blue (thin); Mote 4 in yellow (thin); Mote 5 in red (thick); Mote 6 in green (thick); Mote 7 in blue (thick); Mote 8 in yellow (thick); Mote 9 in gray; The $\times$ are the square limits and the $\bullet$ are Base Stations examples.

method to increase the accuracy.

3.3 Model Used

To perform the simulation 9 different nodes that move within the limited area, were used. This limited area is a square with dimensions of 650×650 where the Base Stations are placed. The Base Stations were equally distributed during the simulations cases, as much as possible, within the square to grant the maximum coverage area possible within the simulation area and this way maximize the efficiency of the expected results. The motes movements path were predefined and can be seen on the Figure 3.3. The same path of the motes was used in all the cases in order to be able to compare always the same movements.

As visible by the map image on Figure 3.3, some of these motes have a more predictable movement while others motes a more random movement. This random movement has been created by a script, which gives higher priority to the forward movement, since is assumed that an animal moves more often forward then backwards. By these variety on the movements there has been created to certain extent a

brought representation for a sensor movement.

Furthermore, as is visible in the map image, the movement of the motes does not have uniform behavior in terms of velocity, meaning that the speed is not the same in all the cases. Moreover, the definition of speed in TinyOS is not clear since the Operating System is event driven, and does not really on predictable CPU cycles. Therefore we use displacement between samples for the mote movement, as further explained on Chapter 4.1, although the displacement gives us the idea of speed during the paths of the different motes.

Our experiments were based on the pure Centroid Model, which has been explained in Chapter 2.3 and implemented according to Formula 2.1. For our case, the Modified Centroid Location method introduces the history of the previous positions, which is included in the formula for the Modified Centroid Location. In our experiments three different computing weights of the historical information have been considered: uniform historical weight, linear historical weight, and exponential historical weight.

This way, to perform the calculation for mote position, the Modified Centroid Method includes the historical information where the following formulas are used. Here $P_i^{'}(x,y)$ indicates the position of unknown node i given by its two dimensional coordinates. The calculated positions using the pure Centroid where calculated using the Formula 2.1 and are given by $P_h(x,y)$, where h are previous calculated positions, starting from the oldest position (meaning $P_0(x,y)$ is the oldest position and $P_h(x,y)$ is the last position calculated). This way the more recent positions have higher weights.

Based on the previous, the formula for the case when the historical information is used with uniform weight, is given by:

$$P_i^{'}(x,y) = \frac{1}{h+1}\sum_{k=0}^{h} P_k(x,y) \tag{3.1}$$

In this case the all the previous historical positions have the same weight and even the same as the last computed position.

For the case when the historical information is used with linear weight, the formula becomes the following:

$$P_i^{'}(x,y) = \frac{1}{\sum_{k=0}^{h}(k+1)}\sum_{k=0}^{h}(k+1)\times P_k(x,y) \tag{3.2}$$

For the linear historical weight, the current computed position has higher weight then the previous and so forth.

In the case when the historical information is used with exponential weight:

$$P_i^{'}(x,y) = \frac{1}{\sum_{k=0}^{h}(k+1)^{k+1}}\sum_{k=0}^{h}(k+1)^{k+1}\times P_k(x,y) \tag{3.3}$$

The exponential historical weight increases even more the weight of the last position over the previous computed one(s).

In all the cases, the node, which is situated on the interception area, computes the position at one single point, based on Centroid Location method and adds to this calculation the historical positions of the mote.

Chapter 4

Experimental Evolution

The simulation to perform our experiments was implemented using TinyOS, with the help of TOSSIM that easily allows debugging, testing, and analyzing the algorithms implemented in a controlled and repeatable environment that allowed extended analysis on the results. As TOSSIM runs on a normal PC, the TinyOS code can be tested using debuggers and other regular development tools, such as python and C++.

In our case python [27] was chosen to interact with TOSSIM, due to its solid and powerful functionality and a relative small quantity of lines of code, which makes it less prone to issues and is easy to debug. This way, the python programing language allowed to compute and process the mote position information for each single iteration step, making it possible to simulate the movement of the motes relatively to the beacons nodes within the closed environment map area as specified in Figure 3.3.

Python is part of the automated simulation implemented and, using the interaction with the TOSSIM, allows the verification if the mote is within the coverage area of the various Base Stations or beacon motes, as required for the Centroid method. This way, triggering the TOSSIM accordingly and for each interaction allowed the mote to calculate its position based on the communication messages exchanged with the Bases Stations and with the historical data to implement our Modified Centroid method. At the end of the simulation the error is calculated by the difference between the real position where the mote is and his computed position, based on the Centroid Modifed Method as described in Chapter 3, using the Formulas 3.1, 3.2 and 3.3, considering uniform, linear and exponential historical weight, respectively.

4.1 Experiments Description

In an initial phase of the tests, during the state of the art verification, a communication protocol has been developed and chosen, between the Base Stations and the beacons. To improve the location accuracy, assuming and keeping in mind the minimal energy waste caused by highly complex computation, has been considered that the Base Stations broadcast their positions, since the Base Stations would have higher power supply, while the moving motes would be powered by batteries. In the simulation model used the Base Stations send multicast (without acknowledge message) messages with their location,

Mote	1	2	3	4	5	6	7	8	9
Maximum Displacement	17.68	17.68	7.07	87.5	24.75	14.14	13.97	12.91	14.02
Minimum Displacement	12.5	12.5	0	0	7.5	1.04	1.04	1.16	1.06
Average Displacement	13.89	12.89	5.83	13.22	17.28	6.42	6.47	6.22	6.6

Table 4.1: Motes displacement to give idea about the mote velocity.

which is received by the motes within their coverage area. In our simulation it has not been introduced any field topology effects resulting in variations on the radio noise, only a simple white noise is considered for communication.

During all the simulations performed, the motes movements have always been the same, as described on the Figure 3.3 from Section 3.3. The tests performed were based on variations of the Base Stations power, to have different coverage areas, and/or Base Stations density in map (the number of the Base Stations), this way makes possible to understand the effect of these changes (power and density of the Base Stations) on the error rate of motes calculated positions using the pure Centroid and the Modified Centroid algorithms when comparing to their real mote position.

To make the motes movement as much random as possible, a script has been developed that generates a pseudo random movement with higher probability for the forward movement, since normally the animals have higher probability to move forward. In Figure 3.3 can be seen that some of the motes have different velocity for their movements and some of then even have a almost random movement (where the backward movement has less probability to occur). Since TinyOS is an event driven OS, we consider the different movement displacement (the length measure between each step or iteraction) to give an idea about the mote's velocity. The motes displacement can be seen in high level on the Table 4.1.

The simulations have been executed with different values on the base stations power that covers the an area of 100, 200, 300, 400, and the number of base stations 8, 16, 25, 36, 49, 56, 64, 72, 81, in the square placed at known positions, on the map with 650×650. We have also centroid simulation where it has been included the case when the moving motes also broadcast their position (with the position the mote has calculated for it self, which due to location errors might differ from their real position) and, this way, become beacons also with a power that covers the area of 200. As final it has been simulated the effect of inclusion of the history of the previous 1, 3, 5 positions the motes have calculated, using 3 different weights for the history of the locations. Considering all these possibilities, it gives 504 different test cases for all the 9 motes, 4536 different cases in total to consider, all performed on a automated script to generate the results consolidated on a Excel file and on a database.

Due to the different cases used on the movement, the different locations of the Base Stations, and with the different coverages used some times the beacons can get to "dark" areas, areas where the beacons do not receive any broadcast information from any neighbor (either other beacon or Base Station). These cases are also verified and considered on the script developed, in order to include information on the historical data.

As mentioned, to implement all these cases, we have developed python scripts that make use of the algorithm implemented in NesC (that runs in TinyOS). The simulation developed considered the move-

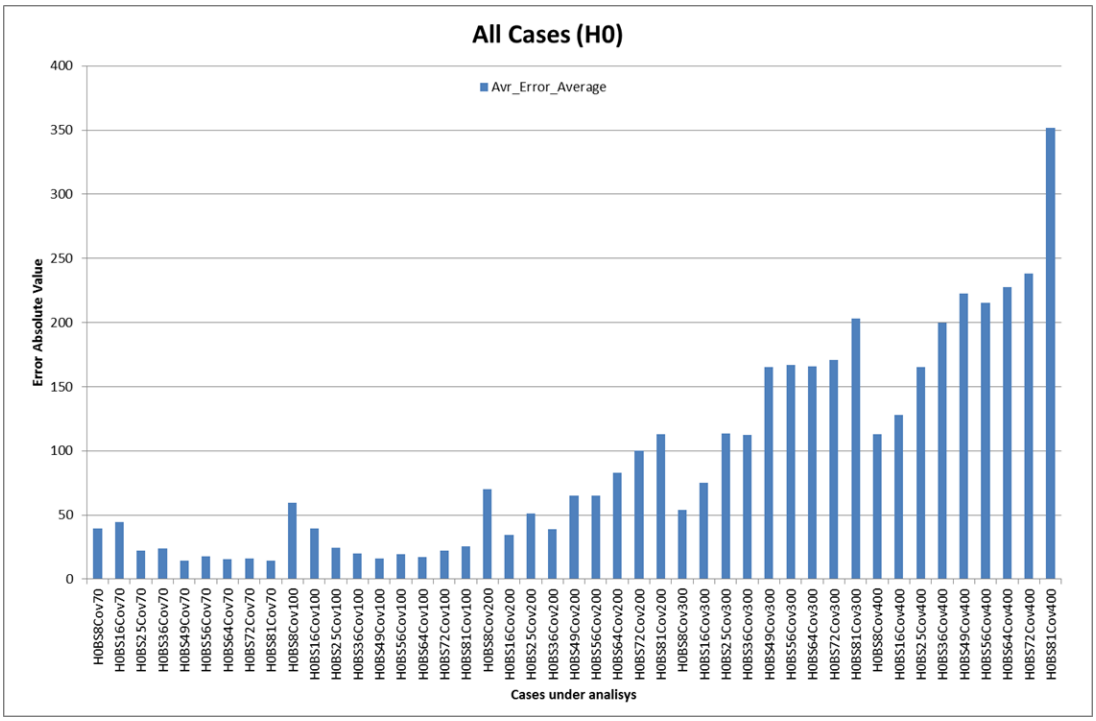


Figure 4.1: Error evolution for all the cases under study, with 8, 16, 25, 36, 49, 56, 64, 72, 81 number of Base Station beacons with 70, 100, 200, 300, 400 Coverage.
*Note the x axis mentions the case considered, where **H0** means no history (pure Centroid Method), the number after **BS** means the number of Base Stations, and the number after **Cov** means the coverage value considered.*

ment of the motes in at least 100 different positions, each of them considered a different interaction of the historical information, and their position retrieved and evaluated in each single one of these positions, by getting the relative position compared to the visible base stations on that particular position using the Modified Centroid methodology as described on the previous chapters. Finally to process all this huge amount of data, it has been implemented with python an automated graphical output generator to easier understand and compare the results, as the ones showed in Appendix A and B.

4.2 Experimental Results

Based on the experiments performed, we can notice that the results have on average a high error, in general more than 10% (error when comparing with the size of the square limited area where the motes are moving).

To clarify the results during the beacons movement, it has been performed verifications for each step of the movement (it can be seen on Appendixes A and B the error of each single step of the movement in detail). These error values have been included on the automated script to generate of the graphics where is expressed the error evolution during the 100 points of the movement.

Graphic on Figure 4.1 presents the error for all the cases studied (8, 16, 25, 36, 49, 56, 64, 72, 81 number of Base Station beacons with 100, 200, 300, 400 Coverage), just to give an high level idea how the error evolves to guide us to focus on more detail the particular case. These cases are pure Centroid Location Method, meaning no historical information has been considered.

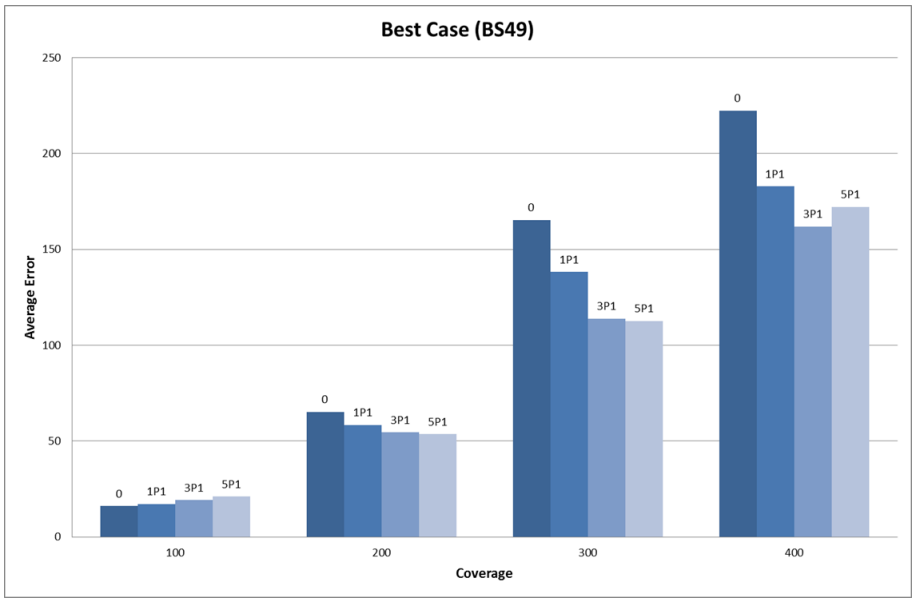


Figure 4.2: Best Case (49 Base Stations) with linear historical effect for different Base Stations coverage.
*Note the **0** means no history (pure Centroid Method), **1P1** means 1 previous historical position, **3P1** means 3 previous historical positions and **5P1** means 5 previous historical position, always using linear weight.*

From the graphic on Figure 4.1 is visible the increase of the Coverage (coverage power the Base Station beacons) increases the error. This error increase is caused by the fact the Centroid Method tends to assume the motes are in the center of the square area, by definition. It is also visible that the increase of the density of Base Station beacons does not always reduces the error on the position calculation, showing that there is a optimal number amount of Base Stations which provides a best result. From the graphic, *the best results occurs when the Coverage is 100, and the density for the number of Base Station beacons is 49.*

Considering the best case for the Centroid method, where the density (the number) of Base Stations is 49 within the square area and Coverage of the Base Stations is 100, further analysis has been focused in more detail this scenarios with different perspectives.

At first, it can be seen the effect of our modification of the Centroid Location Method where is taken in consideration linear weighted history information of the previous steps positions. Here we have considered 1, 3, and 5 previous steps and those are mentioned as 1P1 - 1 previous historical step, 3P1 - 3 previous historical steps and 5P1 - 5 previous historical steps. This evaluation can be seen on the graphic of Figure 4.2.

In this case, the error reduces for higher coverage values on the Base Station beacons and increases for lower coverage because, since in the pure Centroid Method the motes tend to get to the center of the square with higher Coverage of the beacon, by introduction the historical information of previous positions on their calculation, the error is reduced.

The introduction of the intercommunication between the moving motes, makes all the nodes to become beacons, reducing the error when the coverage of the Base Stations is higher, because the mote calculates a position more distant than the center of the square (where it tends to be in the pure Centroid Location). At the same time, if the coverage of the Base Stations has similar level has the coverage of

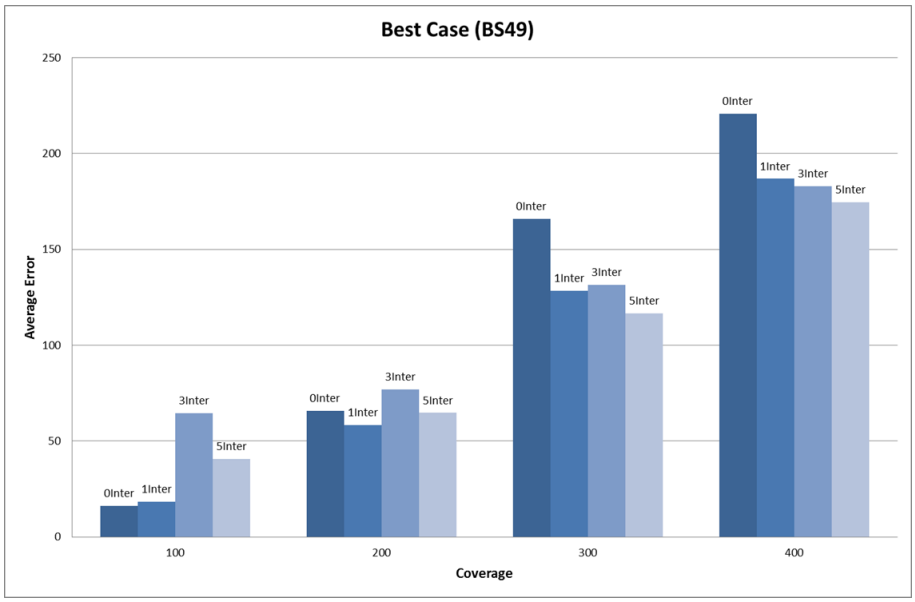


Figure 4.3: Best case (49 Base Stations) with historical effect when intercommunication between mote is introduced, for different Base Stations coverage.
*Note the **0Inter** means no history (pure Centroid Method), **1Inter** means 1 previous historical position, **3Inter** means 3 previous historical positions and **5Inter** means 5 previous historical position, always using linear weight.*

the (also moving) modes, the error is increased by their negative influence, and get even more if the history of the previous positions is also taken in consideration, since it increases the cumulative error, as it is seen by the graphic on Figure 4.3.

Looking to these two cases from the opposite perspective (checking the evolution of the coverage), as expected the error increases with the increase of the coverage. As already mentioned the centroid method intends to positions the motes on the center of the map, but the introduction of the historical and intercommunication reduces the error level as is seen on graphics on Figures 4.4 and 4.5, respectively.

A different perspective is the evaluation that can be seen when the coverage of the Base Station beacons is fixed to 100, but the density (the number) of Base Station density varies, this effect can be seen on the graphic of Figure 4.6. This case shows there is a optimal value for the amount of base stations that provides the best result. This effect happens because the pure Centroid Location methodology makes the motes to be in the center of the visible beacons and, therefore, from one hand, too few Base Stations beacons provides a huge distance in average, since it tends to increases the distance between the beacons and the mote itself; and on the other hand, too many Base Stations beacons in number tends to make the mote to be considered, again, in center of the square.

Based on all this analysis, the optimal case happens in the case the number of Base Stations is 49. More or less around this case same optimal amount of 36, 49, 56 Base Stations is the optimal value of density for the other Coverages values of the Base Stations.

By introducing intercommunication between the moving motes, making all the moving mote also to become beacons, increases the error, this effect is visible on graphic of Figure 4.7 and the error gets even worst if the historical information is also considered. This negative impact is explained because it increases the propagation of the error over the next position steps.

Although by introducing weighted historical information the error does not increases that much, as

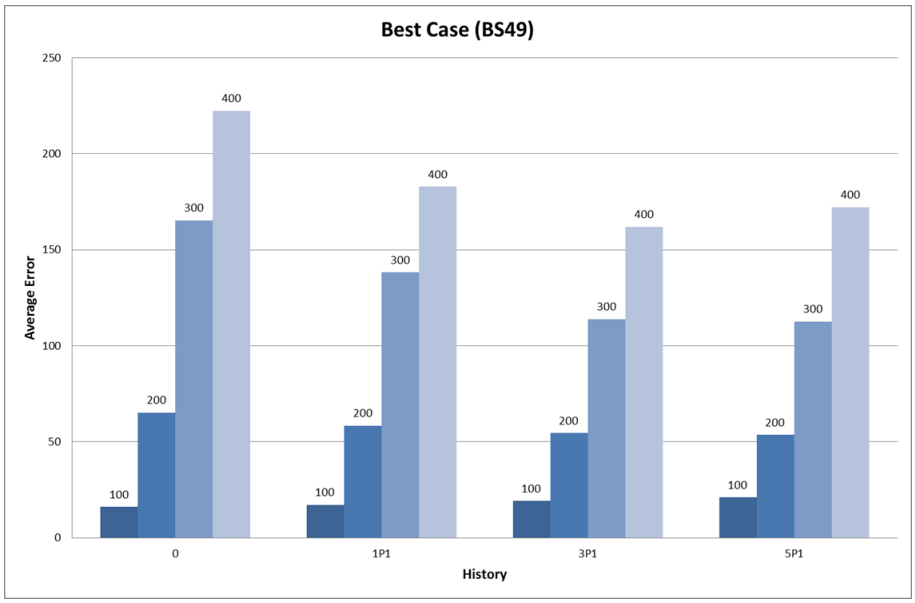


Figure 4.4: Best Case (49 Base Stations) verifying the coverage effect.
*Note the **0** means no history (pure Centroid Method), **1P1** means 1 previous historical position, **3P1** means 3 previous historical positions, **5P1** means 5 previous historical position, always using linear weight, the values 100, 200, 300, 400 are the different coverages considered.*

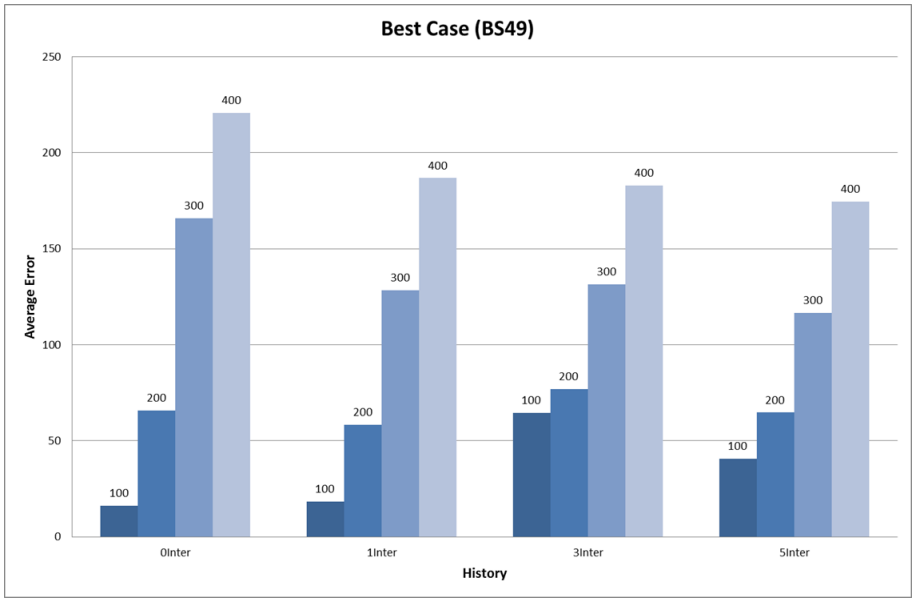


Figure 4.5: Best Case (49 Base Stations) verifying the coverage effect with intercommunication.
*Note the **0Inter** means no history (pure Centroid Method), **1Inter** means 1 previous historical position, **3Inter** means 3 previous historical positions, **5Inter** means 5 previous historical position, always using linear weight, the values 100, 200, 300, 400 are the different coverages considered.*

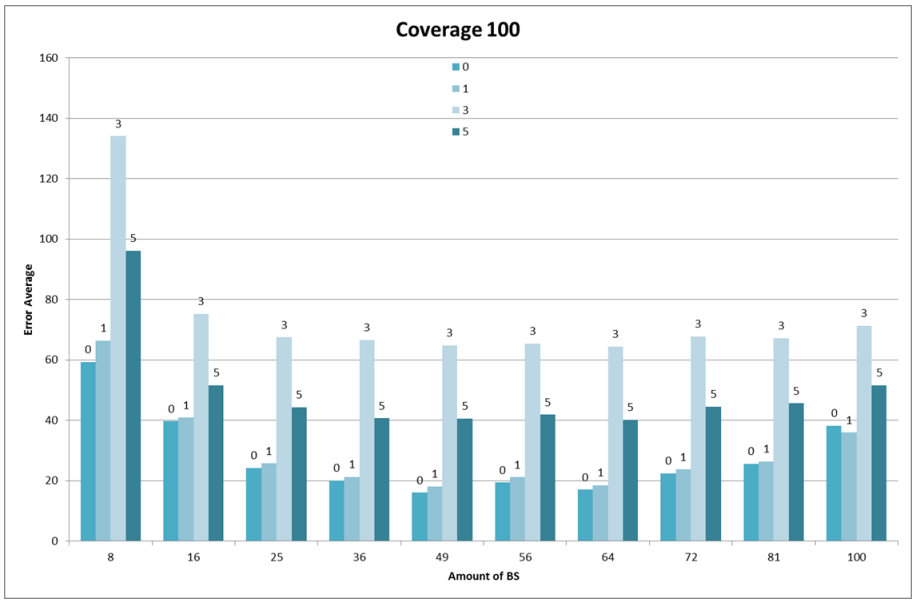


Figure 4.6: Best case (coverage 100) verifying the Base Stations density.
*Note the **0** means no history (pure Centroid Method), **1** means 1 previous historical position, **3** means 3 previous historical positions and **5** means 5 previous historical position, always using uniform weight.*

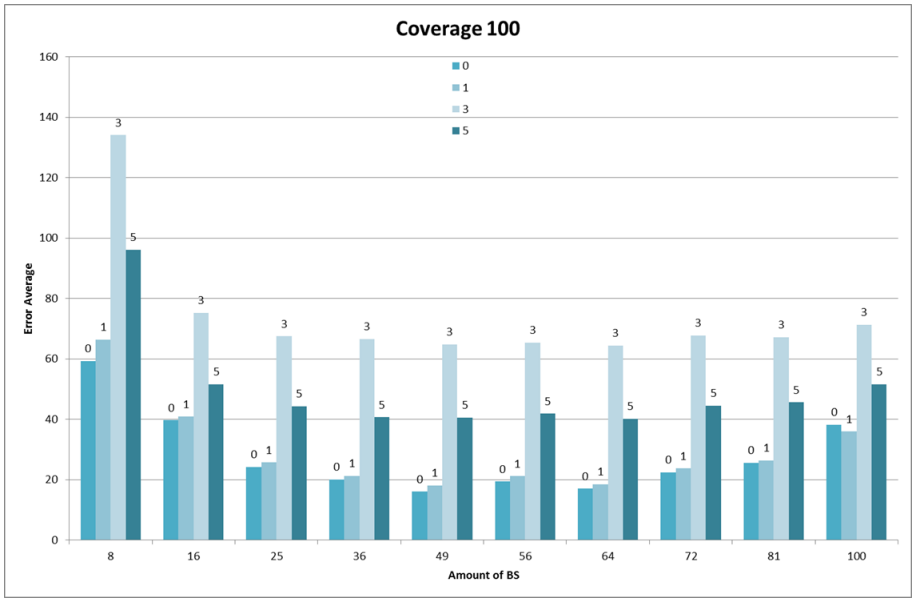


Figure 4.7: Best case (coverage 100) verifying the Base Stations density with intercommunication.
*Note the **0** means no history (pure Centroid Method), **1** means 1 previous historical position, **3** means 3 previous historical positions and **5** means 5 previous historical position, always using uniform weight.*

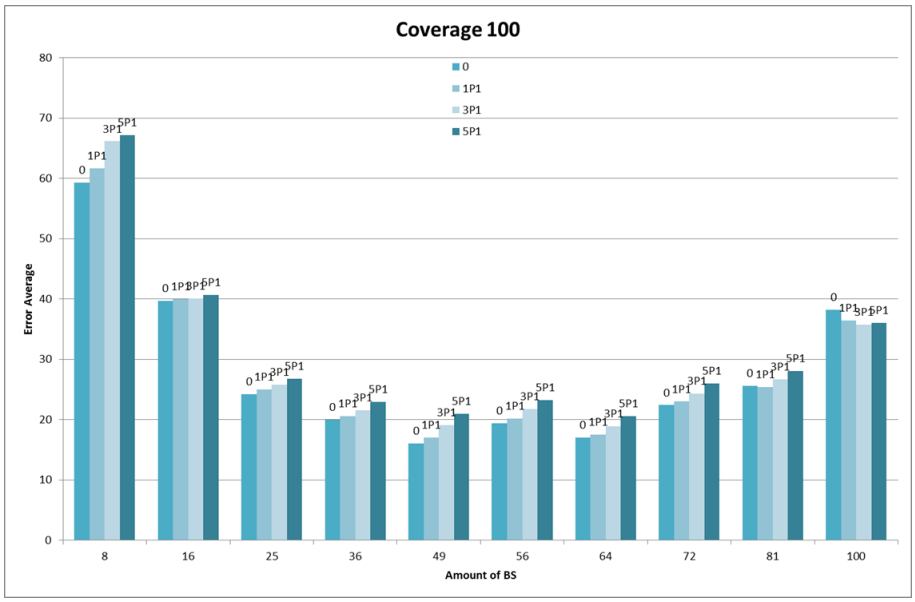


Figure 4.8: Best Case (Coverage 100) effect of the Weighted Historical Information
*Note the **0** means no history (pure Centroid Method), **1P1** means 1 previous historical position, **3P1** means 3 previous historical positions and **5P1** means 5 previous historical position, always using linear weight.*

expected the propagation of the error over the next position steps is reduced by the lower weight of the previous positions, as it can be seen on the graphic of Figure 4.8.

From the examples show before, it is clear the best results comes from the Pure Centroid Method. The introduction of either linear or weighted historical information or even intercommunication between the closest modes has a negative impact on the calculation of their error. The best scenario tested is when using the Coverage 100 on the Base Stations beacons and 49 Base Stations on the density of the number of Base Stations.

Chapter 5

Conclusions

The objective of this thesis was to enhance and improve the pure Centroid Method during the movement of the beacons on a controlled environment, which was partially accomplished as seen in Chapter 4, where the best results for the method used occurs when the Coverage is 100 and the number of Base Stations is 49, on a field of 650×650, and in this case the error is in average 10%.

It got clear on the experiments performed the introduction of historical information increase in certain extent the error and the historical information has a negative impact, therefore from our point of view should not be used in any of the cases experimented. The reason behind is related with the accumulated error that is kept on several samples.

During the analysis stage, several possible parameters were considered, such as the introduction of the RSSI. However, the usage the RSSI was discarded since the RSSI demands a reliable Network Time Protocol (NTP) implemented, which is impossible to guarantee a reliable synchronization when the beacons get to a dark area without coverage from any Base Station. Furthermore, the NTP requires also extra logic on the Base Stations to implement hierarchies, both cases also have negative impact on the power consumptions on the beacons.

5.1 Achievements

During this analysis has been noticed the Centroid Method used on moving elements brings higher error than the Centroid Method used on static elements, this is caused by the fact the Centroid Method cannot be used during several iterations since when a later iteration would be calculated the beacon position had changed already, even thought the error is in average 10%. This error is calculated when comparing with the size of the square limited area, where the motes are moving.

From the experiments performed the minimal errors occurs when the Coverage is 100, and the density for the number of Base Station beacons is 49, in this best scenario case the error has considerable good value and drops to around 2%. During the experiments it must be mentioned the great power of the language chosen, python, which proved to help the experiments performed that allowed to create graphics, excel files, integrations in SQL, in a automated way. Based on all the experiments performed during

the development of this work it was proven the Centroid Method has the best performance when is used alone, without including any other of the experiments, such as introduction of historical information or intercommunication with neighbors.

5.2 Future Work

Since in our believe the major reason of the high level value is caused by the fact the beacons are moving and this way it cannot be implemented several interpolations of the pure Centroid Method, it would be suggested to understand the impact of the speed when compared to the number of samples when implementing more interpolations on the pure Centroid Method, since the pure Centroid Method seams to have good results when the beacons are stopped. An example of a practical usage of this work would be to know the location of cow on the field, since the cows rarely move backwards; a second future possible approach is to include vectorial information on the movement to reduce the error of the results.

Bibliography

[1] J. Monteiro, L. L. de Oliveira, J. B. Martins, and G. Dessbesell. Hardware implementation of a centroid-based localization algorithm for mobile sensor networks. *Symposium on Circuits and Systems*, pages 2829–2832, May 2011.

[2] J. Monteiro, L. L. de Oliveira, J. B. Martins, and G. Dessbesell. Centroidm: a centroid-based localization algorithm for mobile sensor networks. *Symposium on Integrated Circuits and Systems Design*, pages 204–209, September 2010.

[3] N. Bulusu, J. Heidemann, and Estrin. D. gps-less low cost outdoor localization for very small devices. *IEEE PerCom*, 2000.

[4] J. D. Gibson. *The Mobile Communications Handbook*. IEEE PerCom, 1996.

[5] N. Bulusu, J. Heidemann, and D. Estrin. *Density Adaptive Algorithms for Beacon Placement in Wireless Sensor Networks*. IEEE ICDCS, 2001.

[6] O. Hernandez, S. C. V. Jain, and P. Bhargava. Position location monitoring using ieee 802.15.4/zigbee technology. *Freescales Wireless Connectivity Operation*, pages 67–69, 2007.

[7] J. Bachrach and C. Taylor. *"Localization in Sensor Networks," in Handbook of Sensor Networks: Algorithms and Architectures*. I. Stojmenovic, Ed., 2005.

[8] K.-Y. Cheng, K.-S. Lui, and V. Tam. Localization in sensor networks with limited number of anchors and clustered placement. In *Wireless Communications and Networking Conference, 2007*, pages 4425–4429. IEEE WCNC 2007, March 2007.

[9] A. A. Ahmed, H. Shi, and Y. Shang. Sharp: A new approach to relative localization in wireless sensor networks. In *IEEE ICDCS*. IEEE ICDCS 2005, 2005.

[10] M. Maroti, B. Kusy, G. Balogh, P. V. olgyesi, A. Nadas, K. Molnar, S. Dora, and A. Ledeczi. Radio interferometric geolocation. In *3rd International Conference on Embedded Networked Sensor Systems (SenSys)*, San Diego, California, USA, November 2005. SenSys.

[11] N. Patwari and A. O. Hero. Indirect radio interferometric localization via pairwise distances. In *3rd IEEE Workshop on Embedded Networked Sensors (EmNets 2006)*, Boston, MA, May 2006. EmNets.

[12] R. Huang, G. V. Zaruba, and M. Huber. Complexity and error propagation of localization using interferometric ranging. In *IEEE International Conference on Communications ICC 2007*, Glasgow, Scotland, June 2007. IEEE.

[13] N. A. Alsindi, K. Pahlavan, and B. Alavi. An error propagation aware algorithm for precise cooperative indoor localization. In *IEEE Military Communications Conference MILCOM 2006*, Washington, DC, October 2006. IEEE.

[14] E. Elnahrawy, X. Li, and R. P. Martin. The limits of localization using signal strength: A comparative study. In *IEEE SECON*, Santa Clara, California, October 2004. IEEE.

[15] J. Krumm, L. Williams, and S. G. Smartmovex on a graph - an inexpensive active badge tracker. *UbiComp*, 2002.

[16] T. Christ and P. Godwin. A prison guard duress alarm location system. *IEEE ICCST*, 1993.

[17] A. Smailagic, J. Small, and D. P. Siewiorek. Determining user lcation for context aware computing through the use of a wireless lan infrastructure. *IEEE*, 2000.

[18] M. Youssef, A. Agrawala, and A. U. Shankar. Wlan location determination via clustering and probability distributions. *IEEE PerCom*, 2003.

[19] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann. Weighted centroid localization in zigbee-based sensor networks. In *IEEE International Symposium on Intelligent Signal Processing*, Madrid, October 2007. WISP 2007, IEEE PerCom.

[20] C. Alippi and G. Vanini. A rssi-based and calibrated centralized localization technique for wireless sensor networks. In *IEEE International Conference on Pervasive Computing and Communications Workshops*, Pisa, March 2006. PERCOMW 2006, IEEE.

[21] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. *SenSys*, November 2005.

[22] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. *INFOCOM*, June 2002.

[23] T. Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. *SenSys*, November 2003.

[24] T. Instruments. 2.4 ghz ieee 802.15.4 / zigbee-ready rf transceiver.

[25] Tinyos: Operating system design for wireless sensor networks. http://www.tinyos.net.

[26] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesc language: A holistic approach to networked embedded systems. In *Programming Language Design and Implementation (PLDI)*, June 2003.

[27] Python is a multi-paradigm programming language. https://www.python.org.

Appendix A

Results in detail, with 49 Base Stations and coverage 100

Here are presented the graphical evolution in detail, on all the cases studied, of the error during the 100 steps for each mote of the 9 motes movements considered, in the case there are 49 Base Stations distributed on the map with coverage 100. For the map presented the case with 49 Base Stations is one of the cases with the lowest error results, as described during analysis of the thesis.

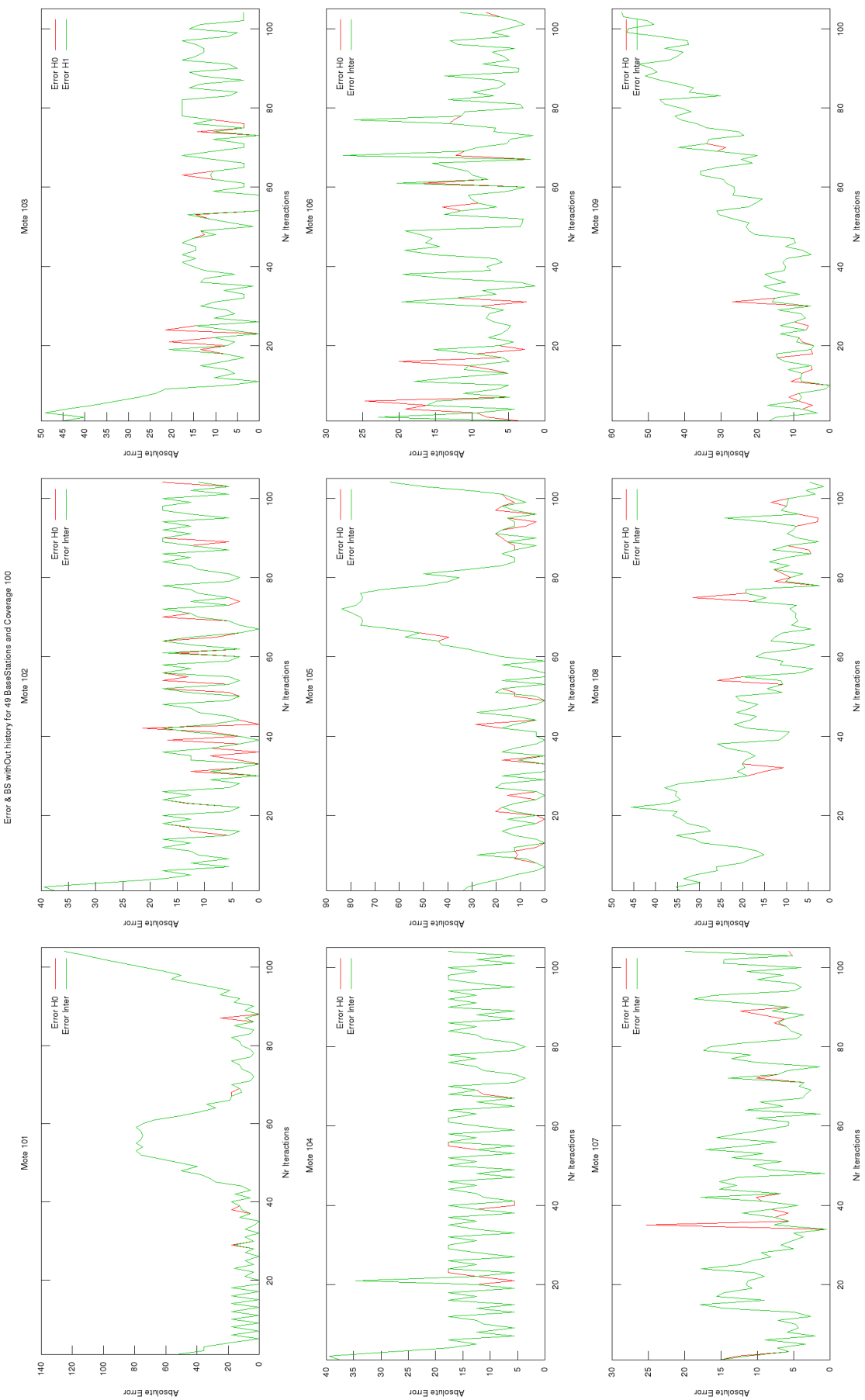


Figure A.1: 49 Base Stations, coverage 100 without historical information.

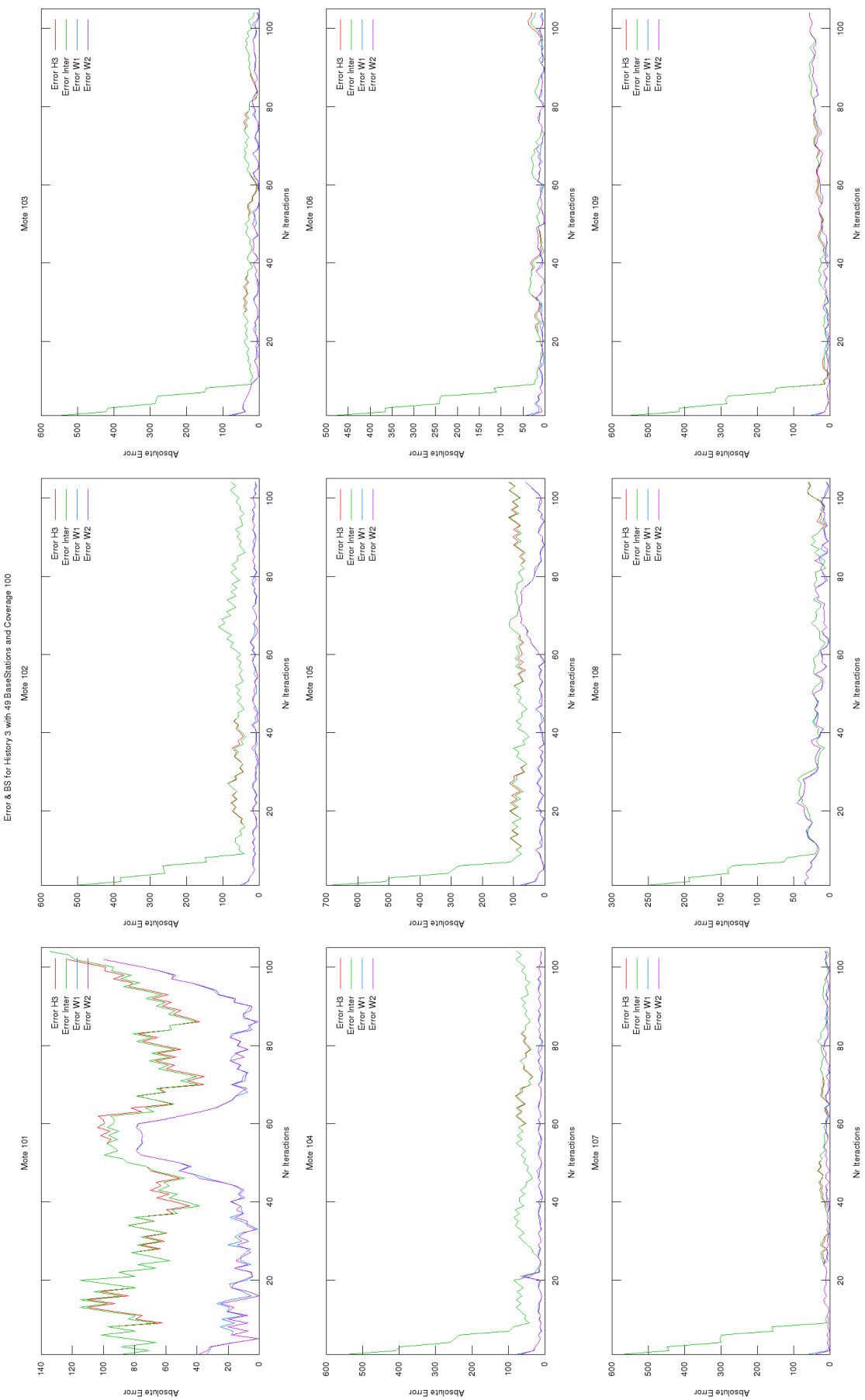


Figure A.2: 49 Base Stations, coverage 100 with 3 previous steps on uniform historical information.

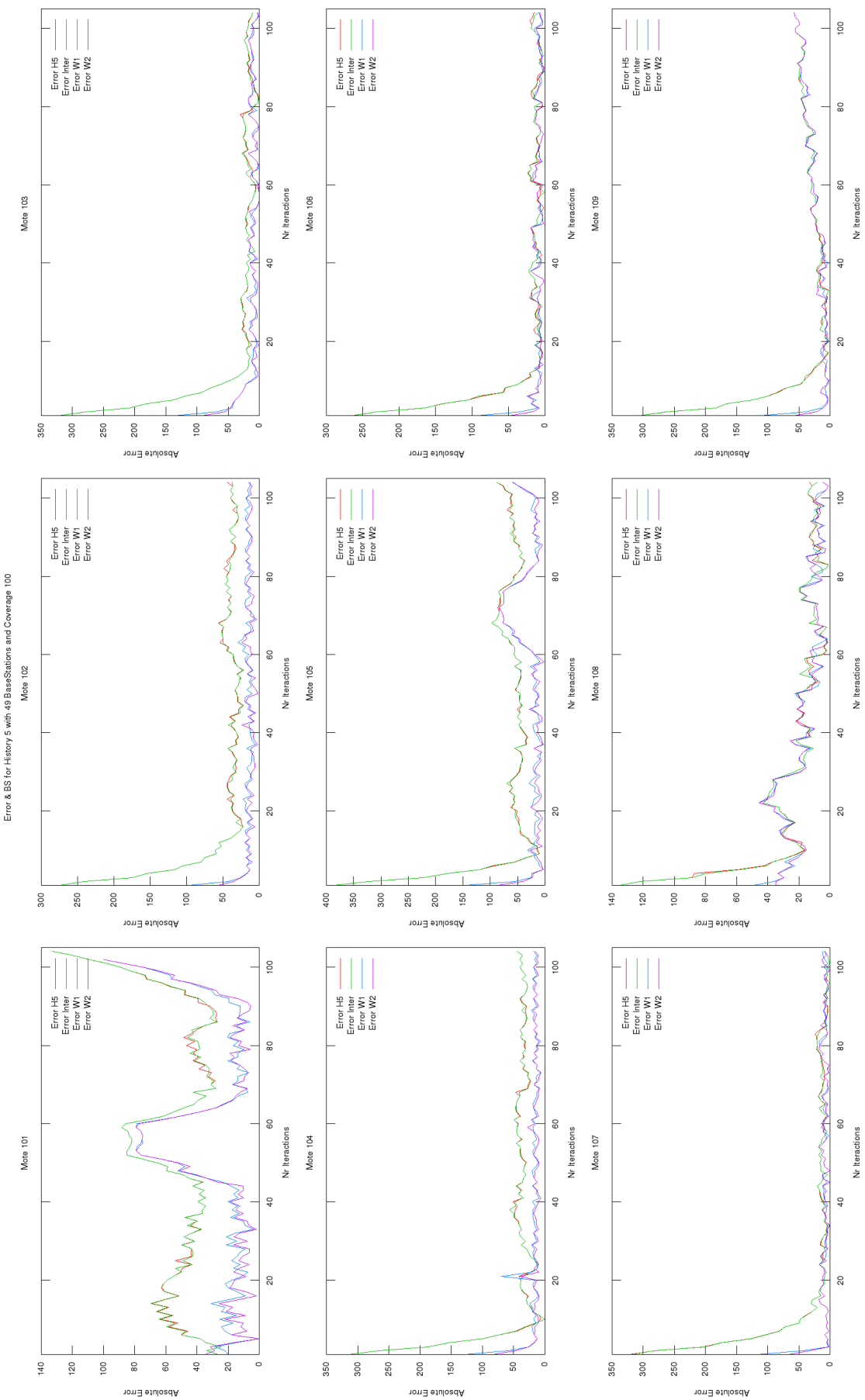


Figure A.3: 49 Base Stations, coverage 100 with 5 previous steps on uniform historical information.

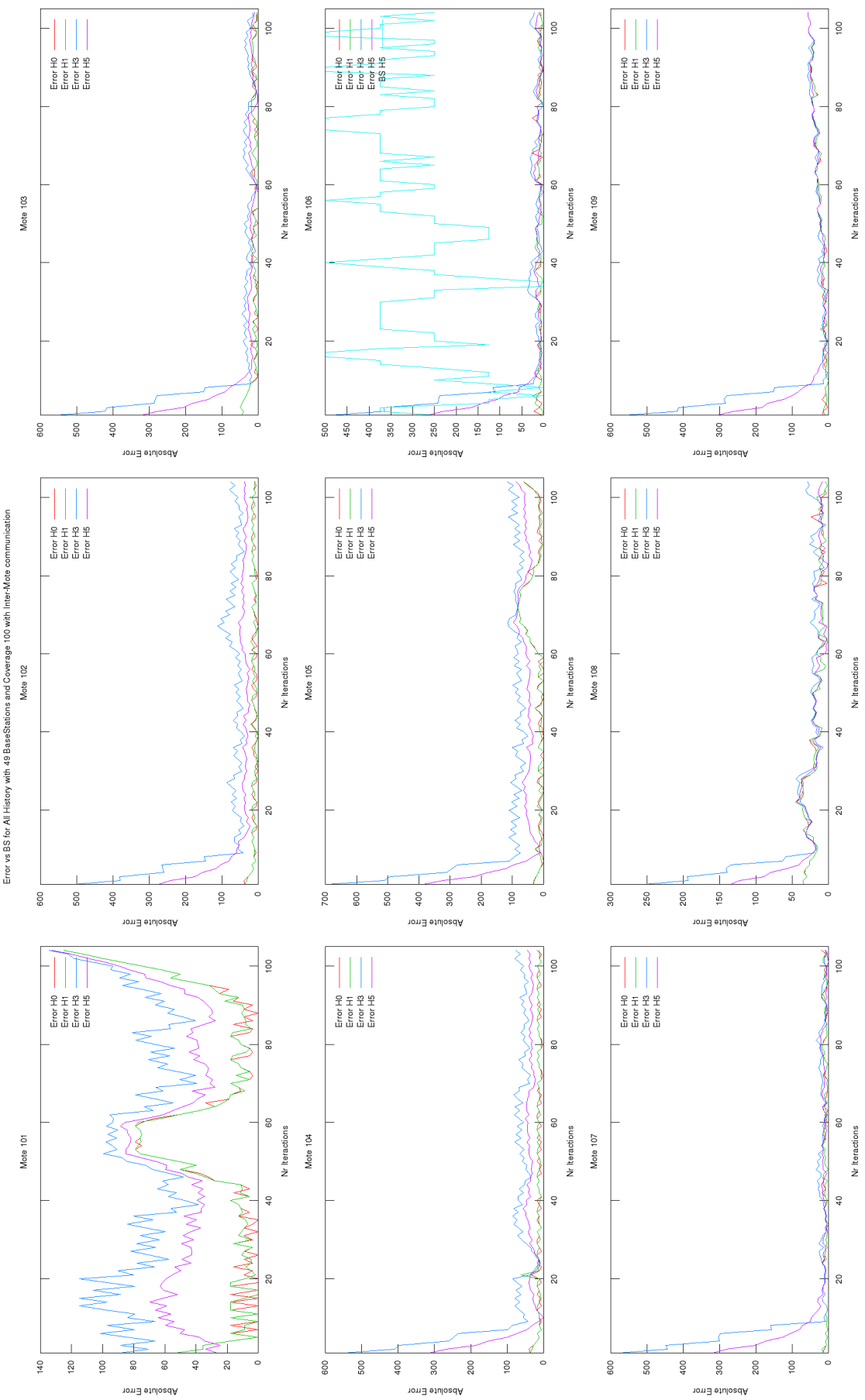


Figure A.4: 49 Base Stations, coverage 100 without historical information, with intercommunication between motes.

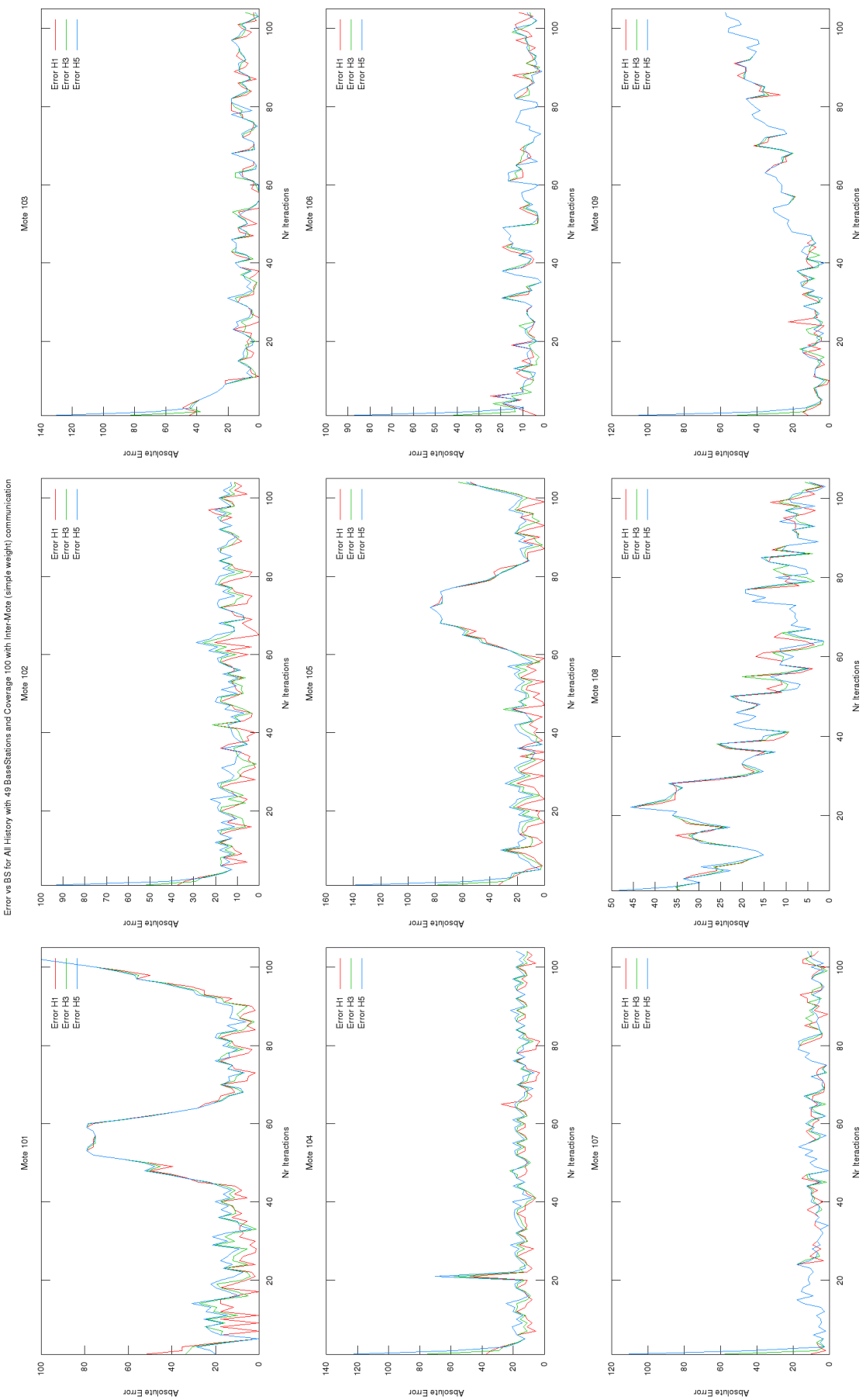


Figure A.5: 49 Base Stations, coverage 100 with 5 previous steps on linear historical information.

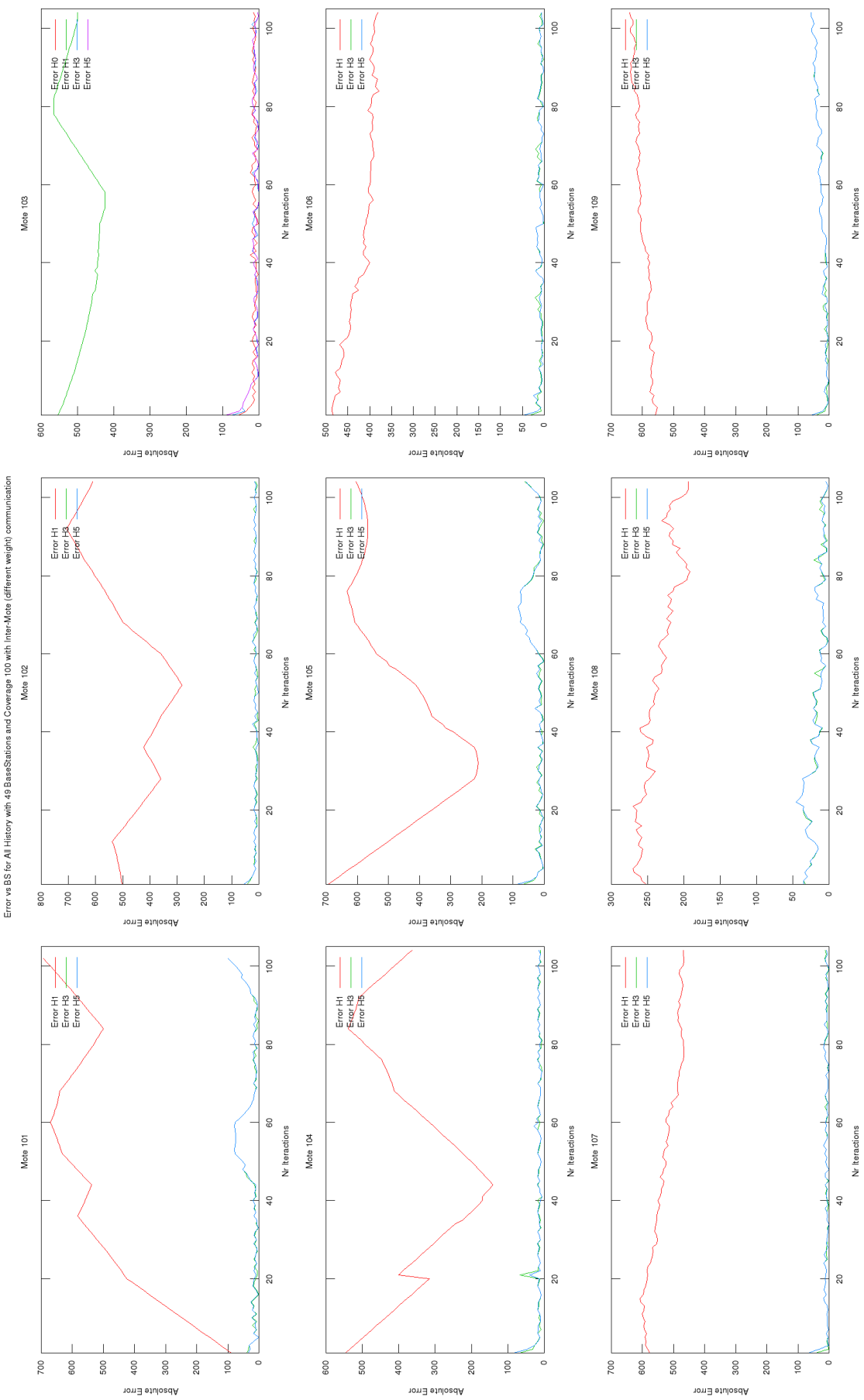


Figure A.6: 49 Base Stations, coverage 100 with 5 previous steps on exponential historical information.

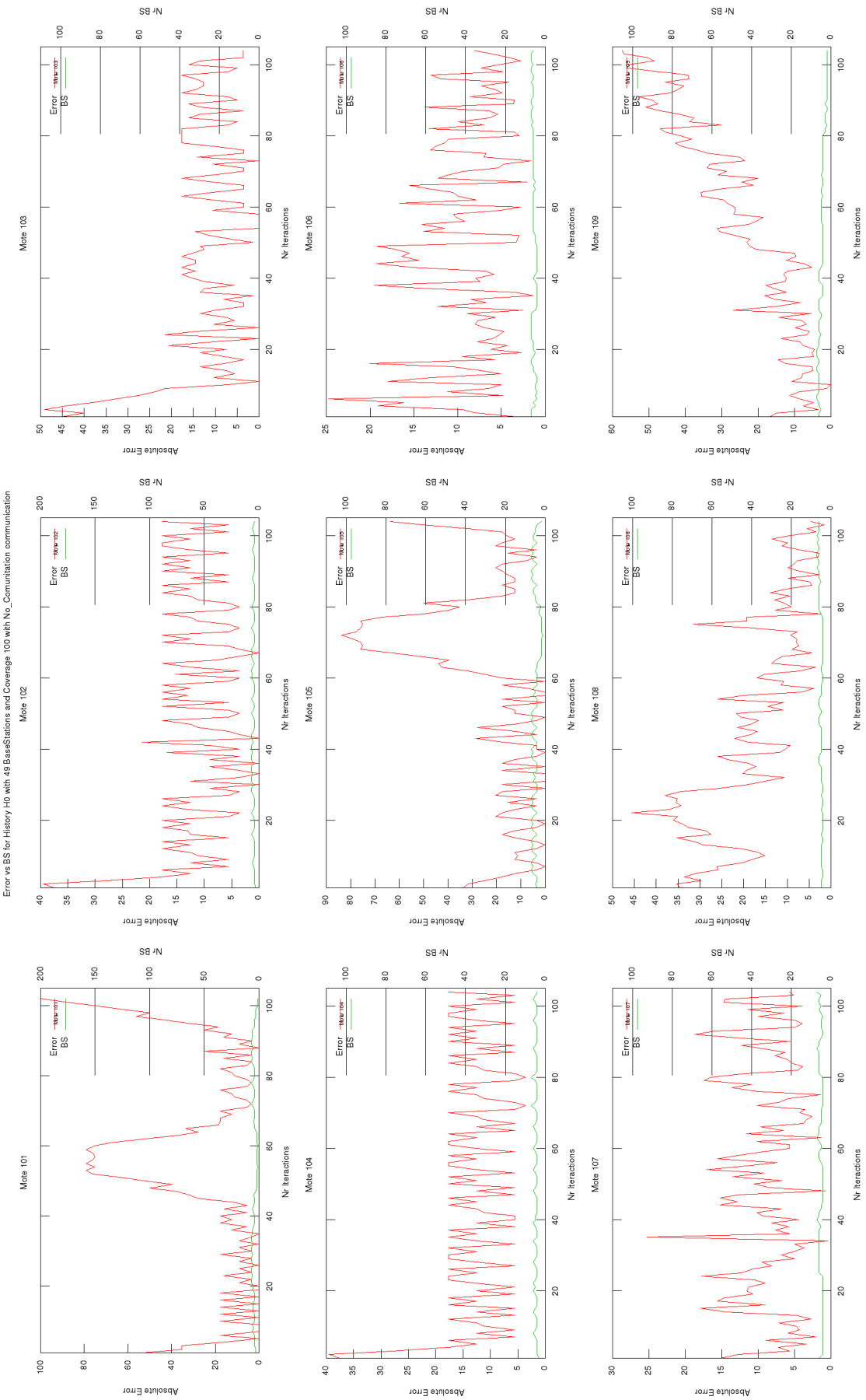


Figure A.7: Pure Centroid with 49 Base Stations and coverage 100.

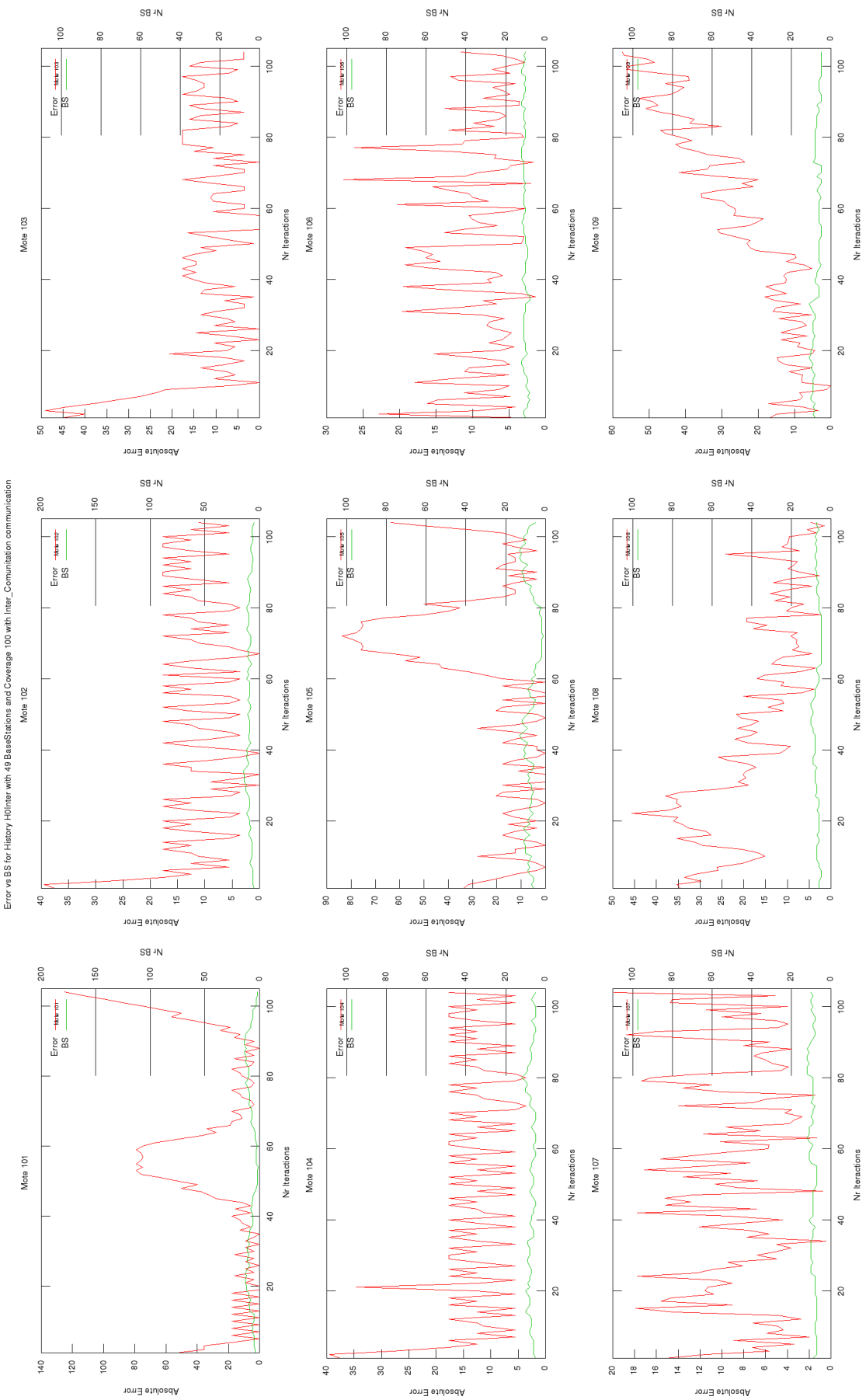


Figure A.8: Pure Centroid with 49 Base Stations and coverage 100, using intercommunication between motes.

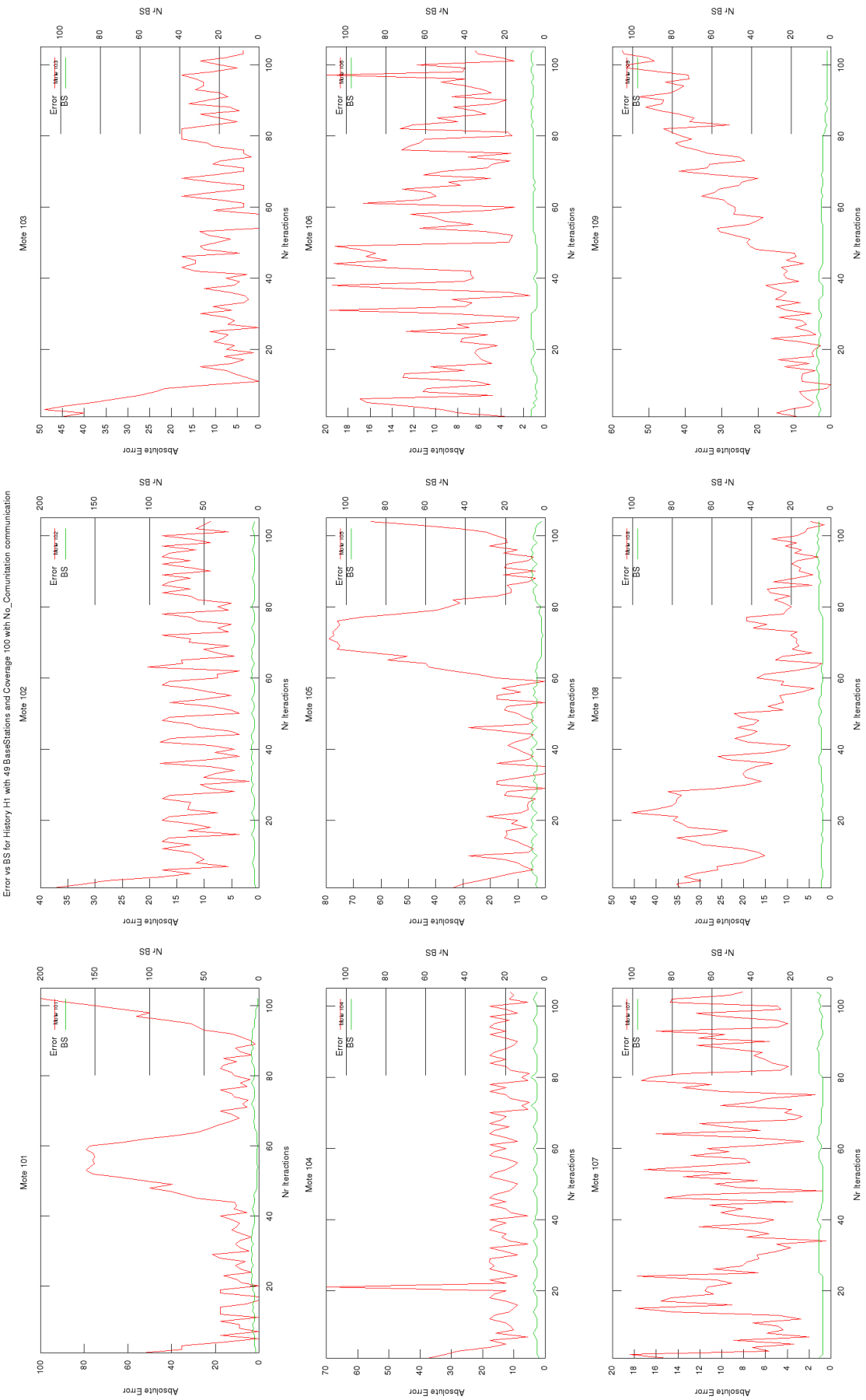


Figure A.9: 49 Base Stations, coverage 100 with last step on uniform historical information.

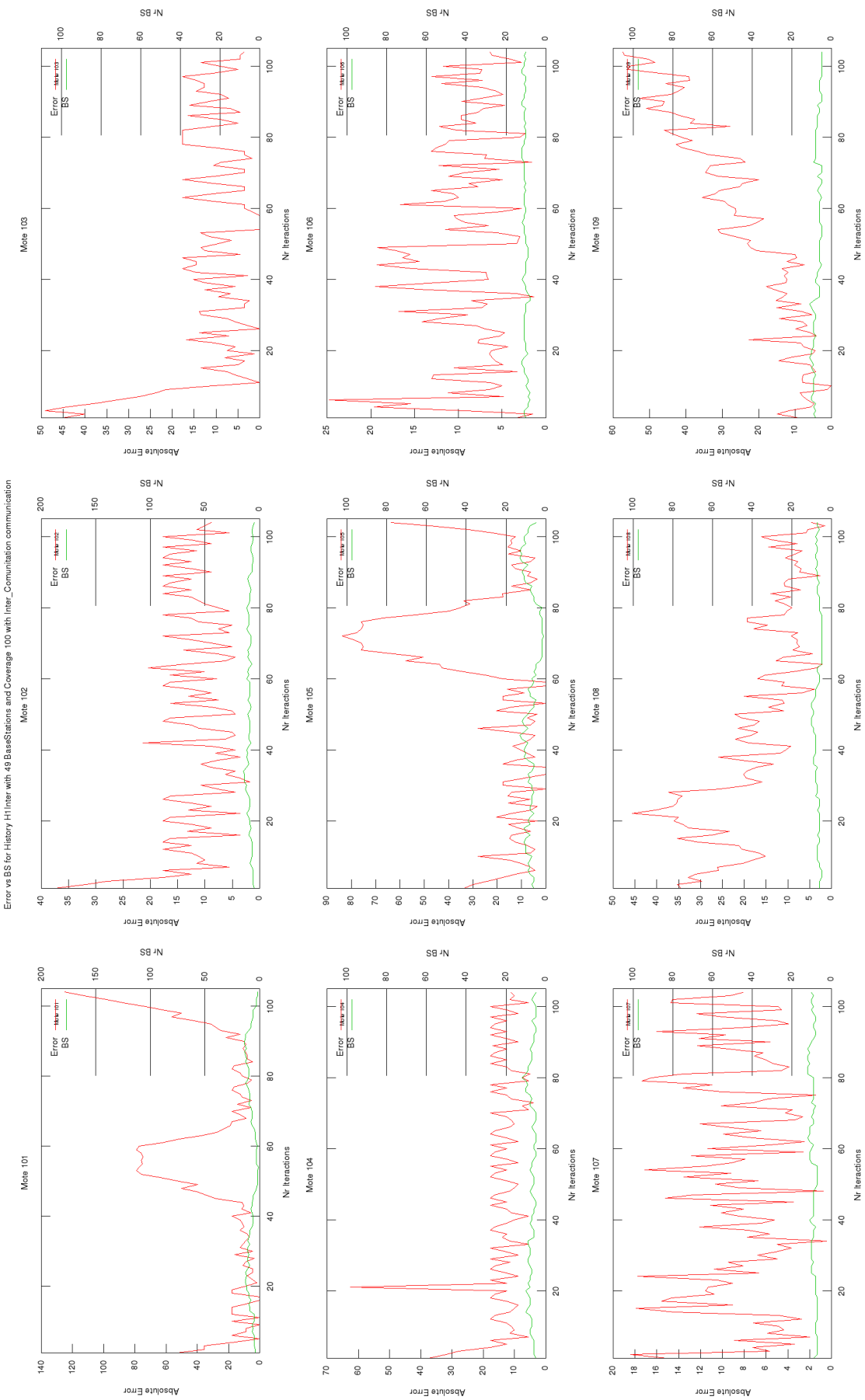


Figure A.10: 49 Base Stations, coverage 100 with last step on uniform historical information, including intercommunication between motes.

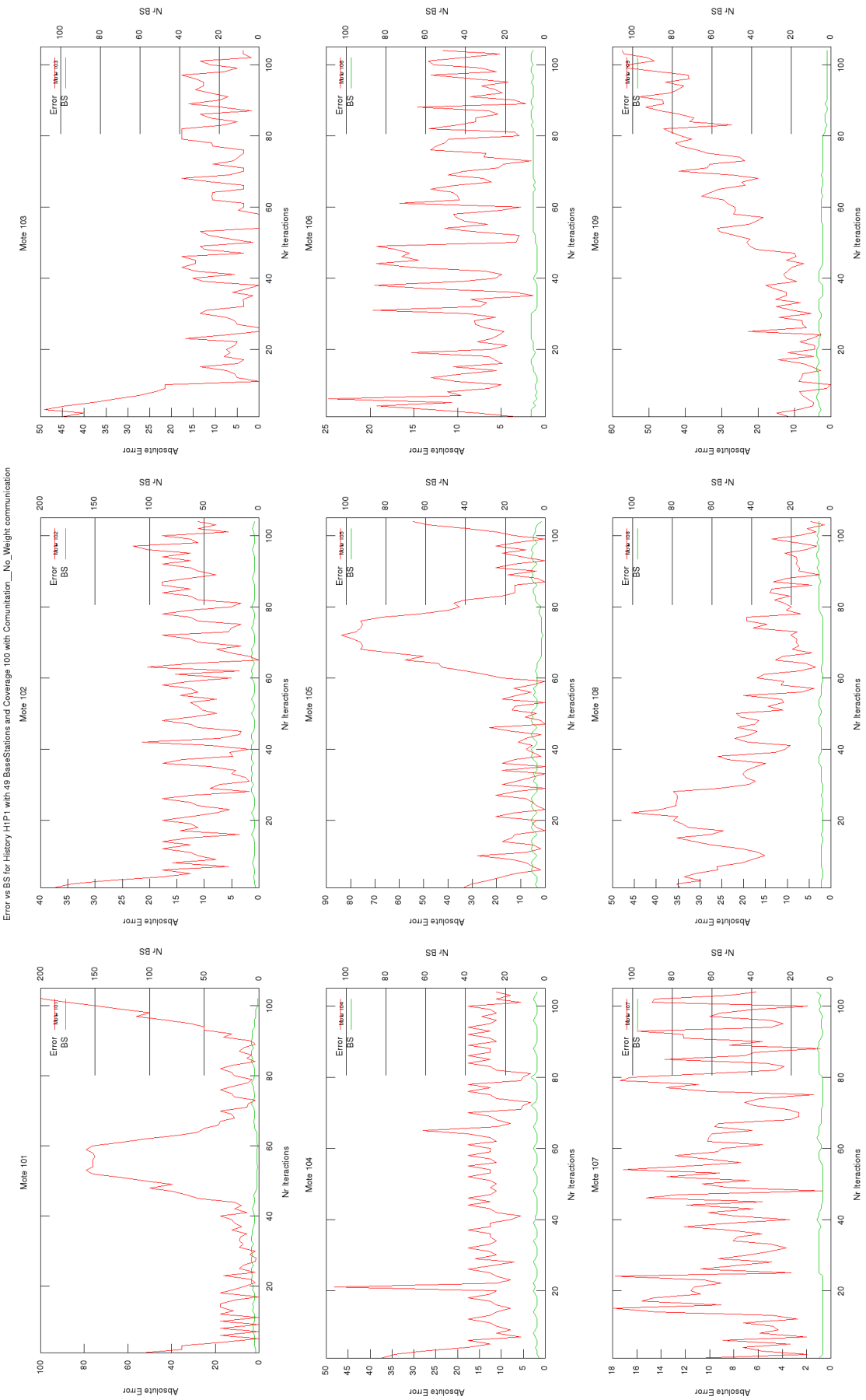


Figure A.11: 49 Base Stations, coverage 100 with last step on linear historical information.

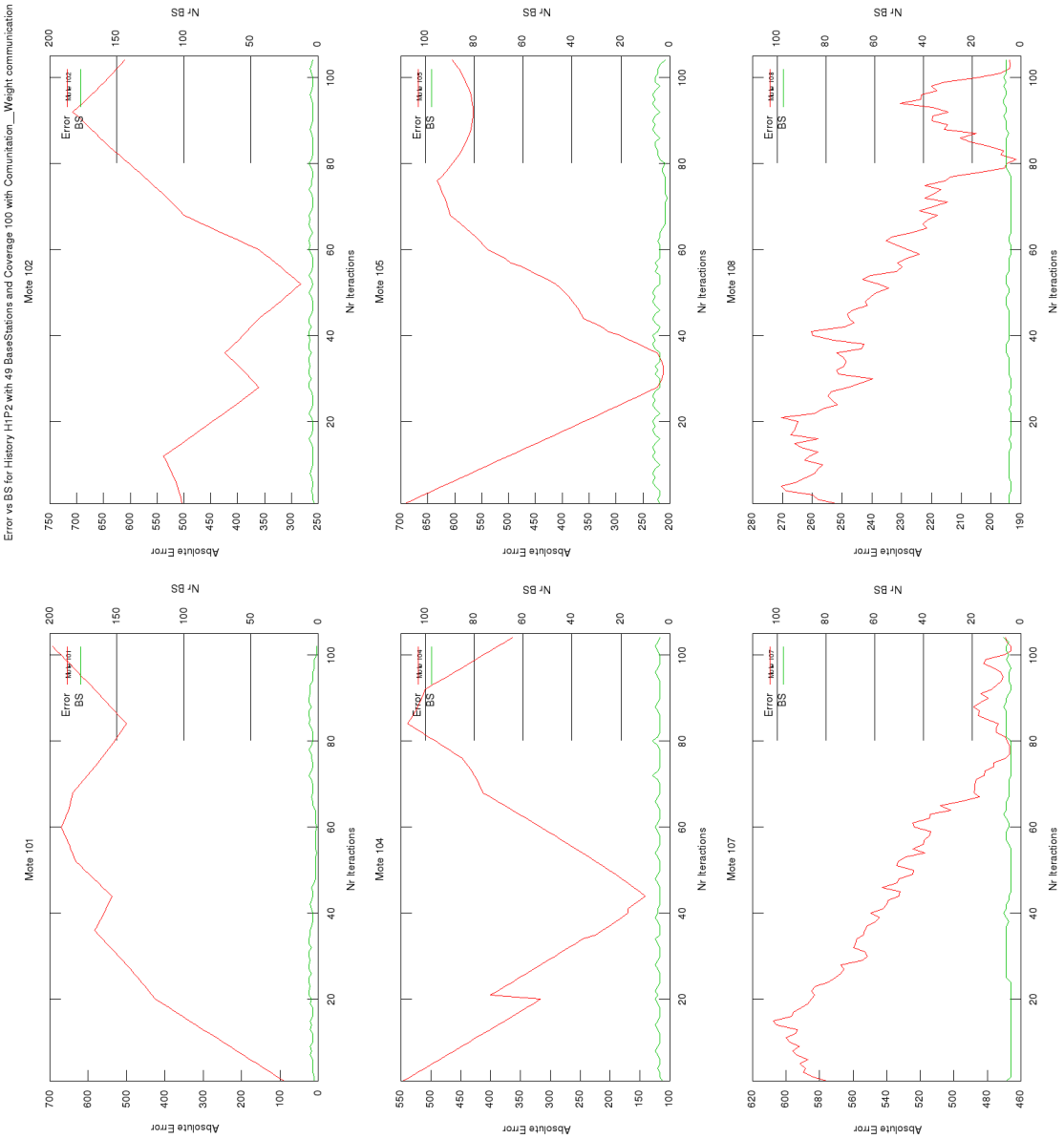


Figure A.12: 49 Base Stations, coverage 100 with last step on exponential historical information.

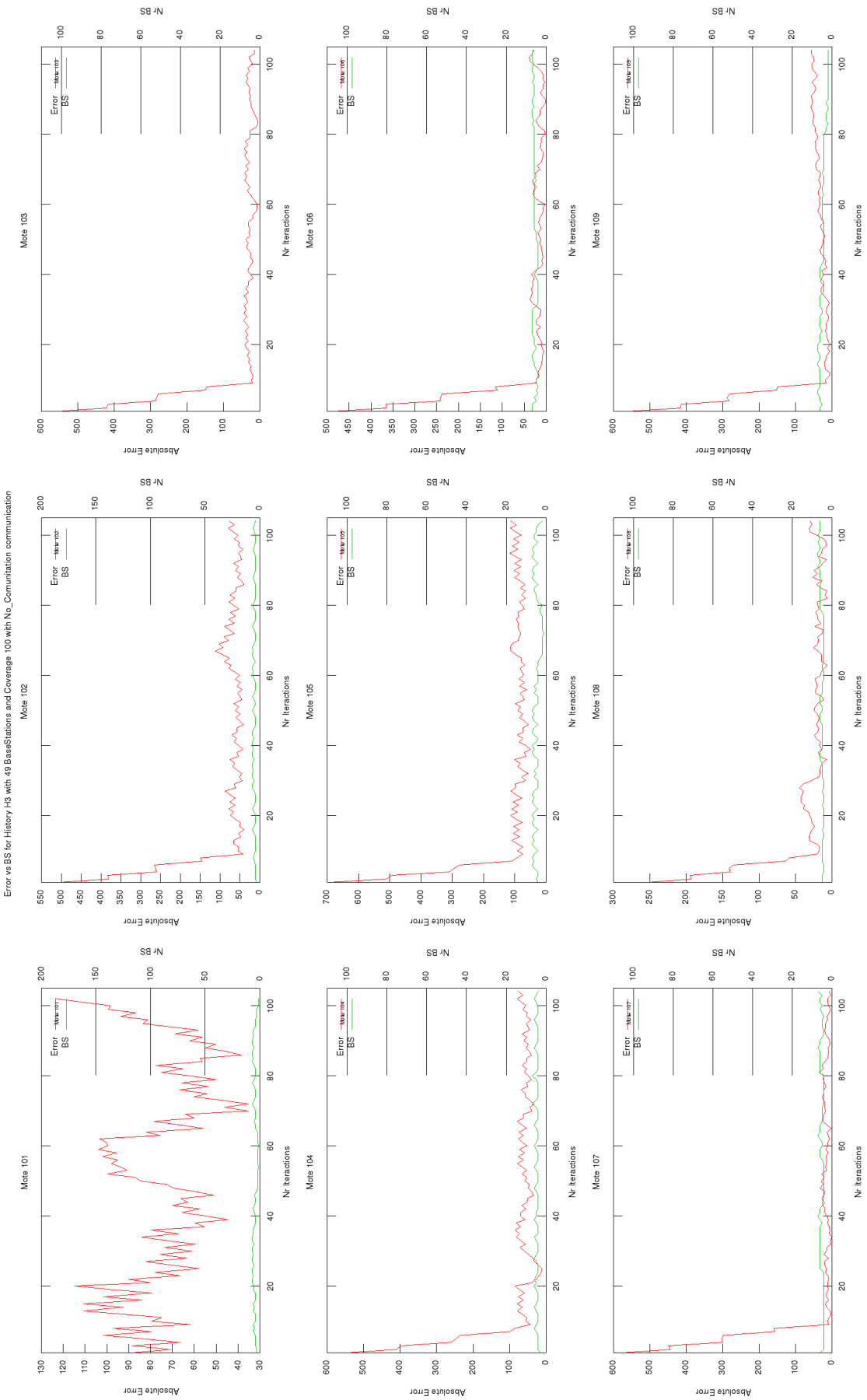


Figure A.13: 49 Base Stations, coverage 100 with previous 3 positions using uniform historical information.

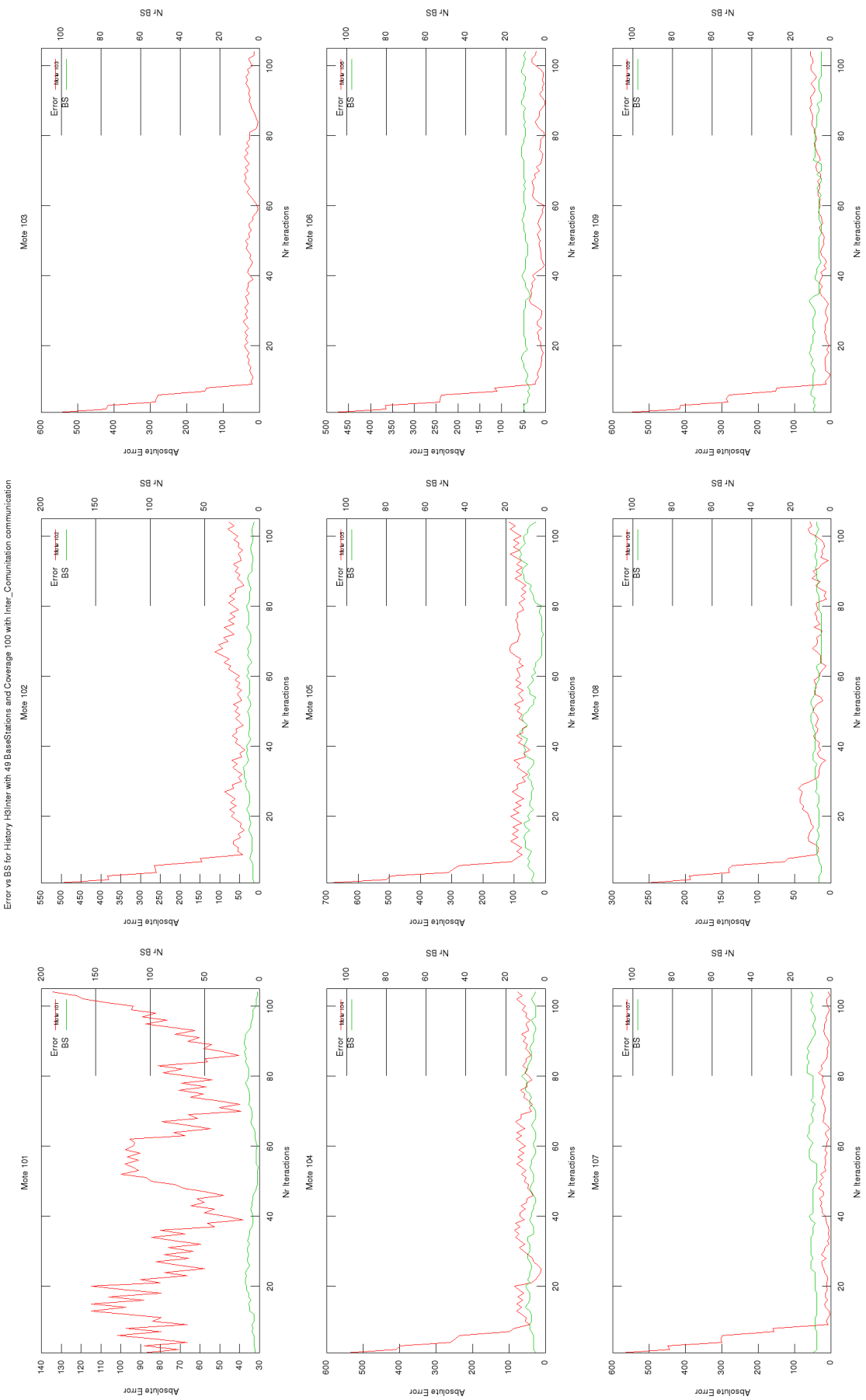


Figure A.14: 49 Base Stations, coverage 100 with previous 3 positions using uniform historical information and intercommunication between motes.

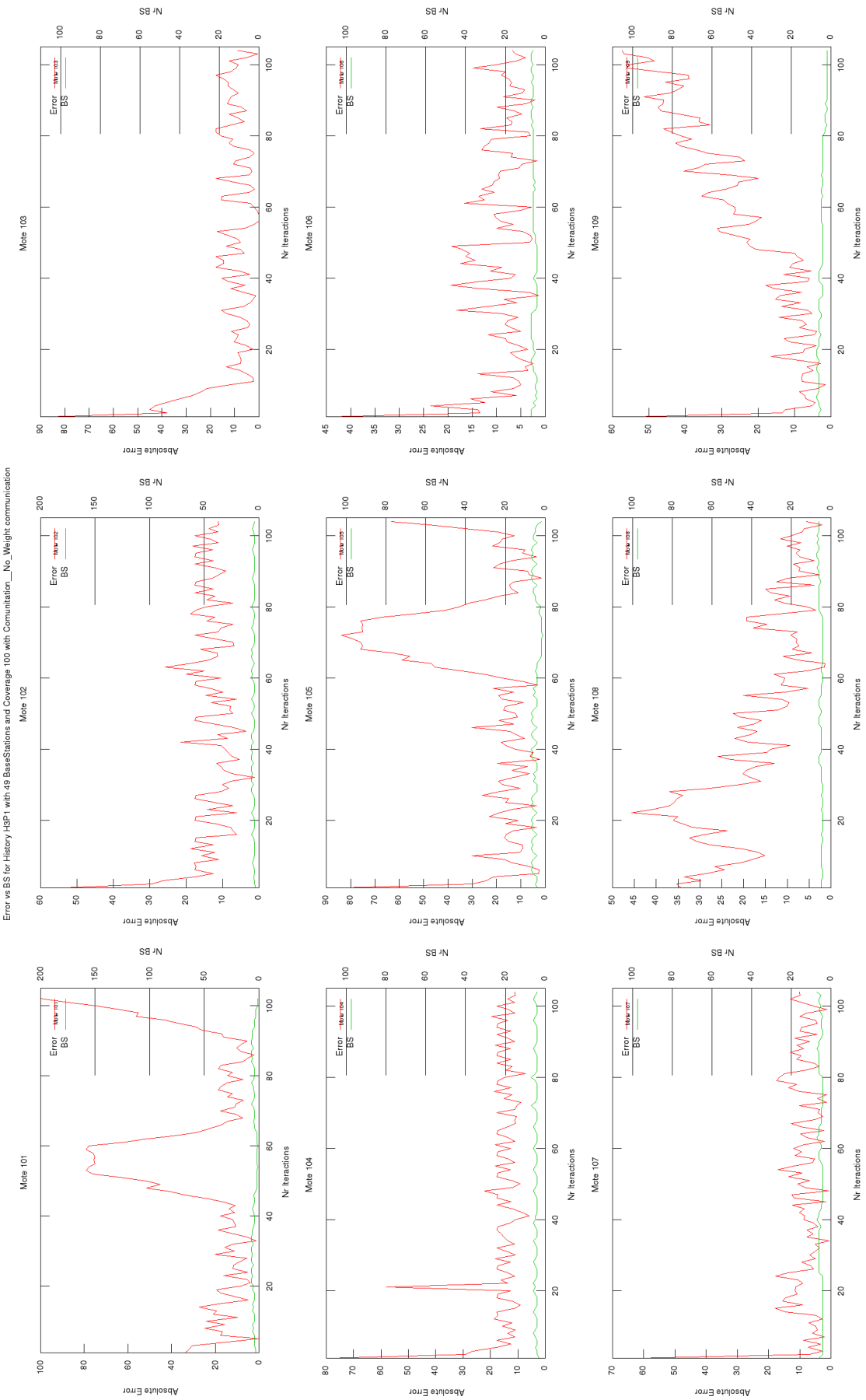


Figure A.15: 49 Base Stations, coverage 100 with previous 3 positions using linear historical information.

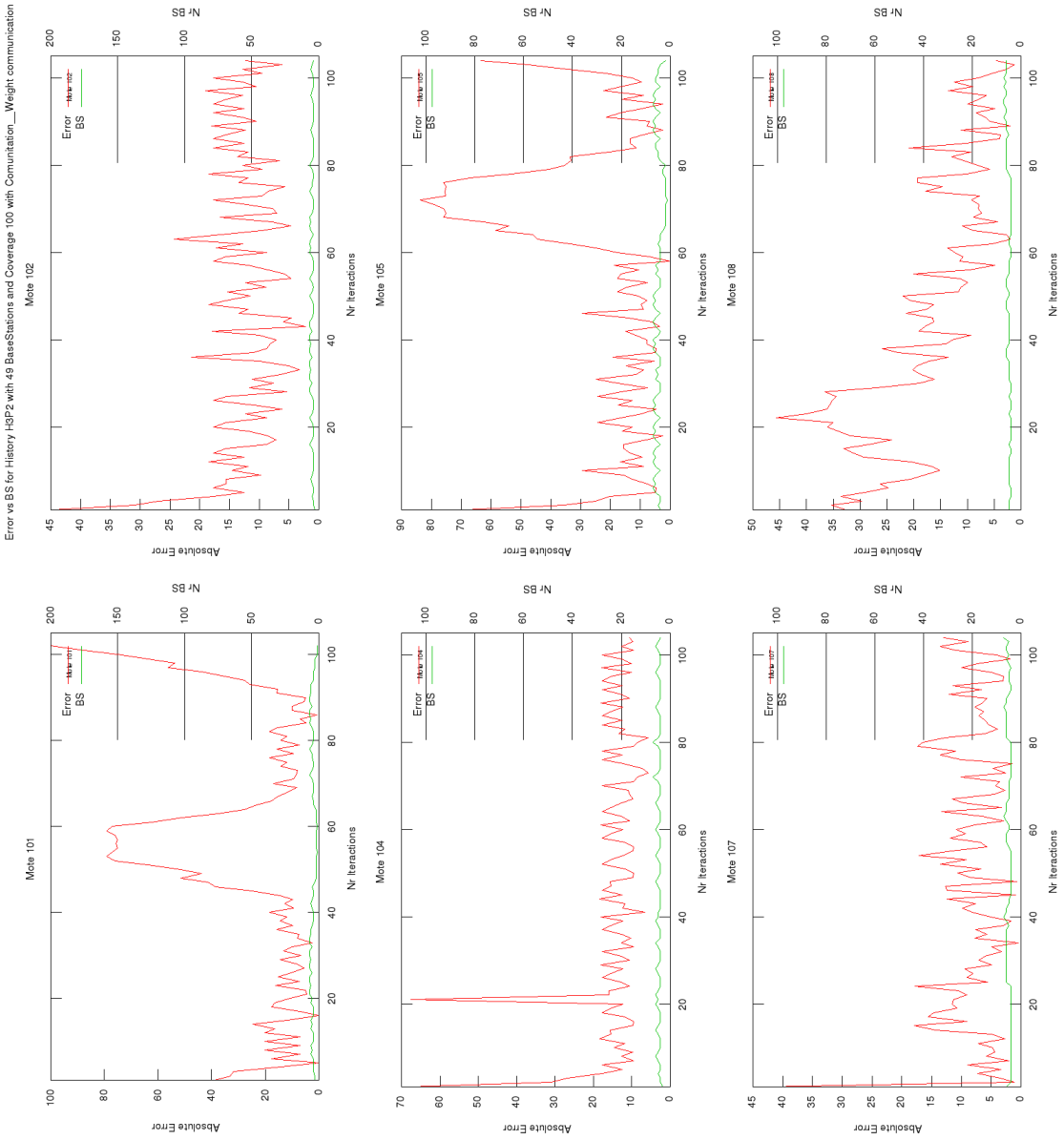


Figure A.16: 49 Base Stations, coverage 100 with previous 3 positions using exponential historical information.

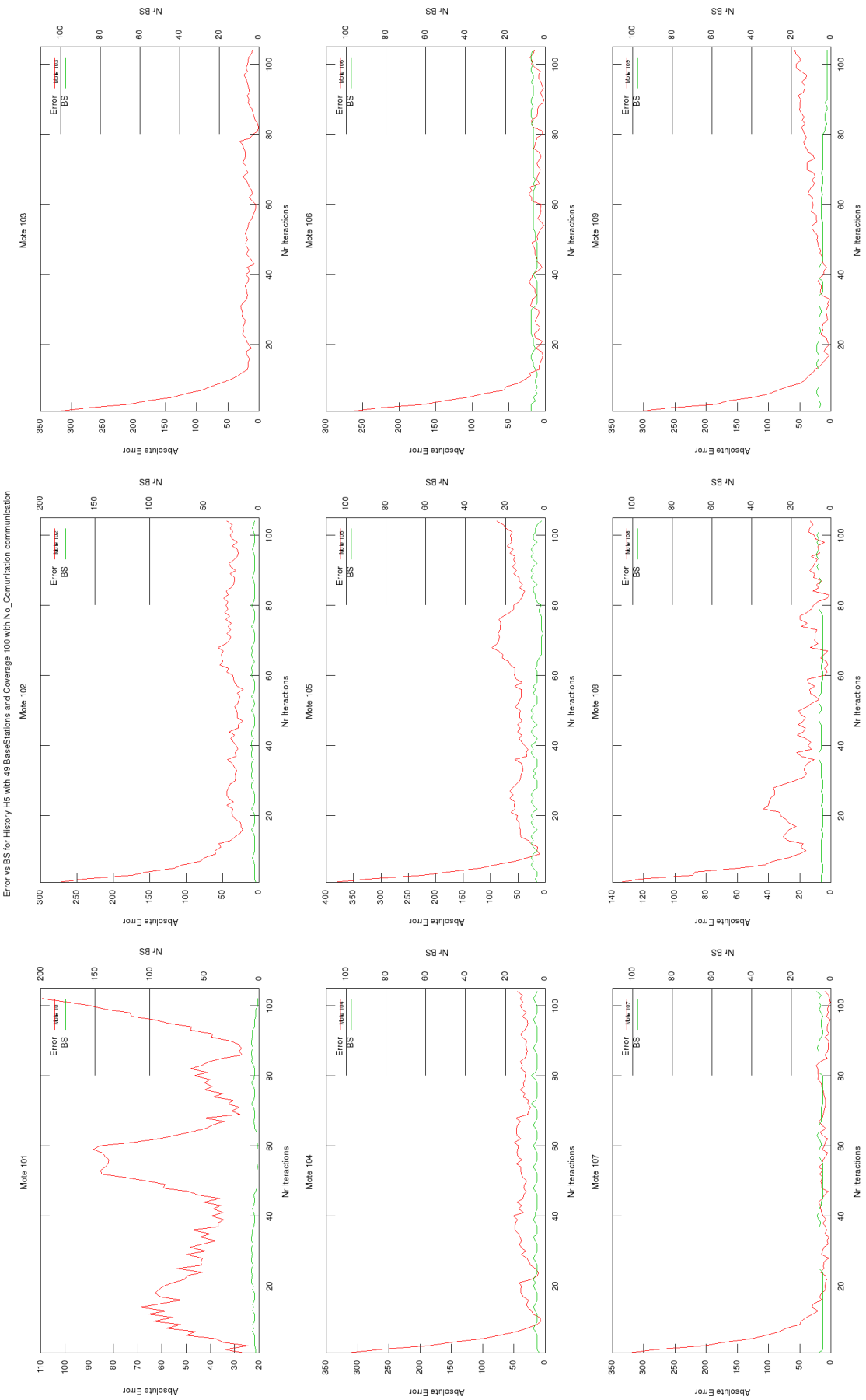


Figure A.17: 49 Base Stations, coverage 100 with previous 5 positions using linear historical information.

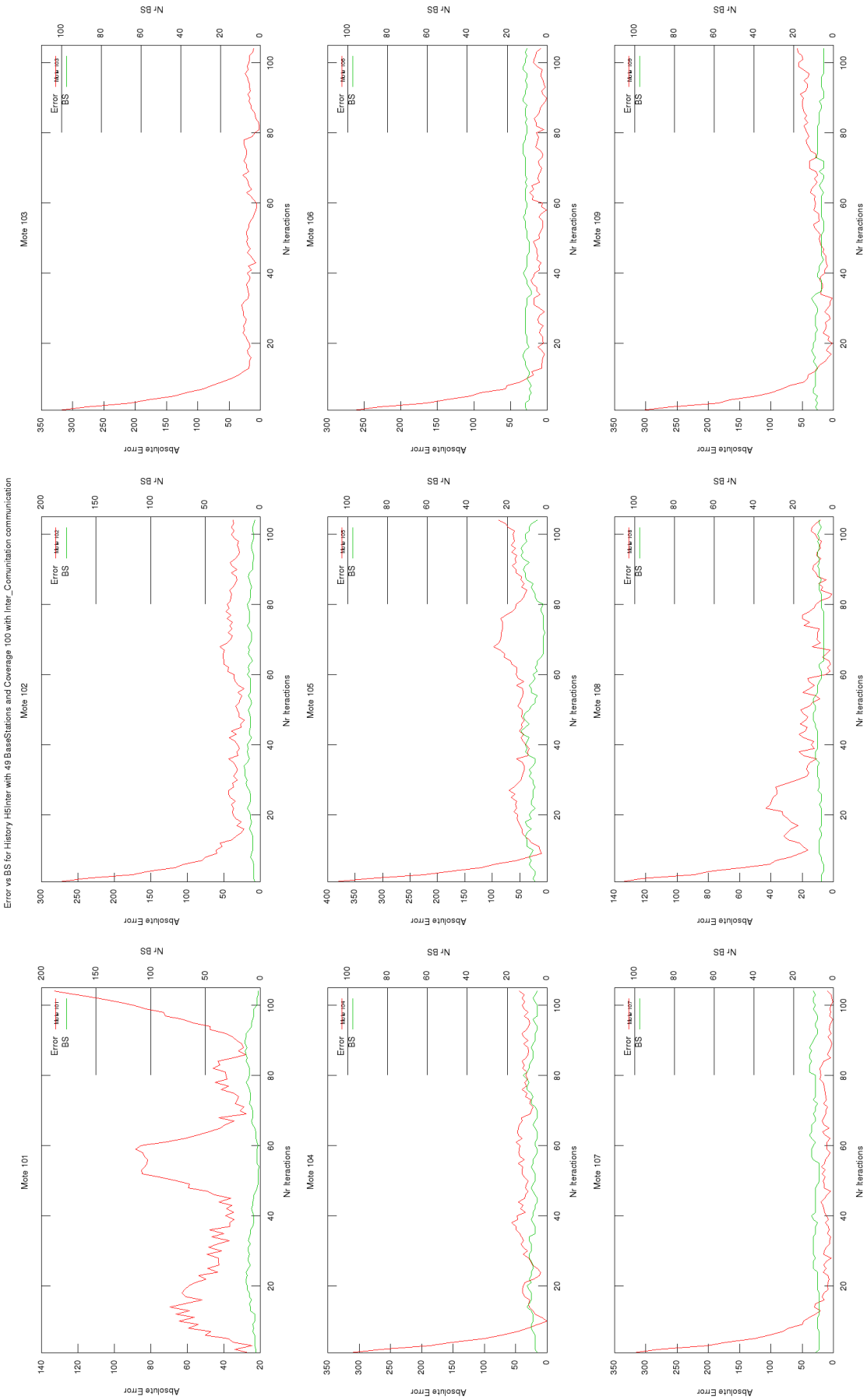


Figure A.18: 49 Base Stations, coverage 100 with previous 5 positions using linear historical information and with intercommunication between motes.

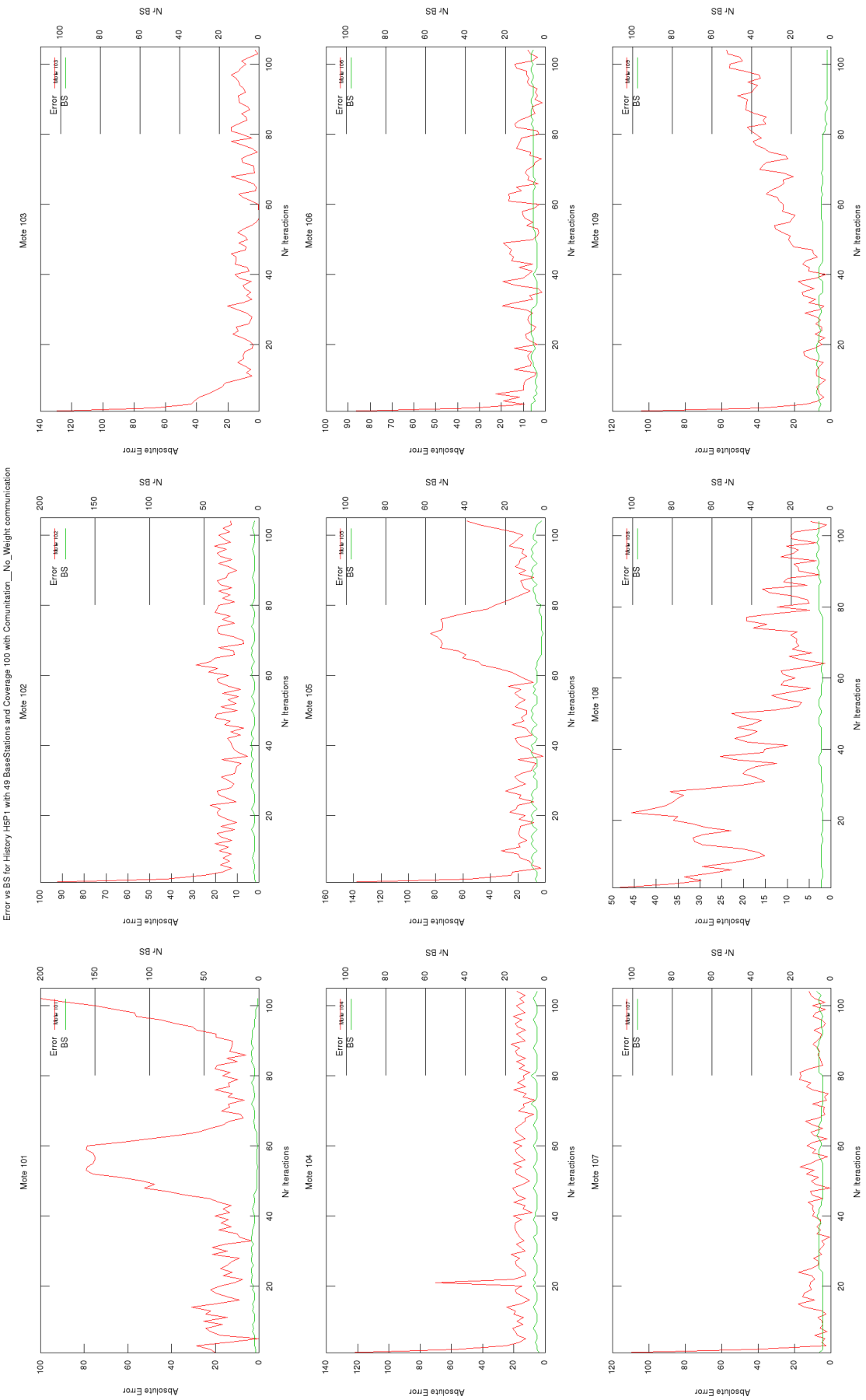


Figure A.19: 49 Base Stations, coverage 100 with previous 5 positions using linear historical information.

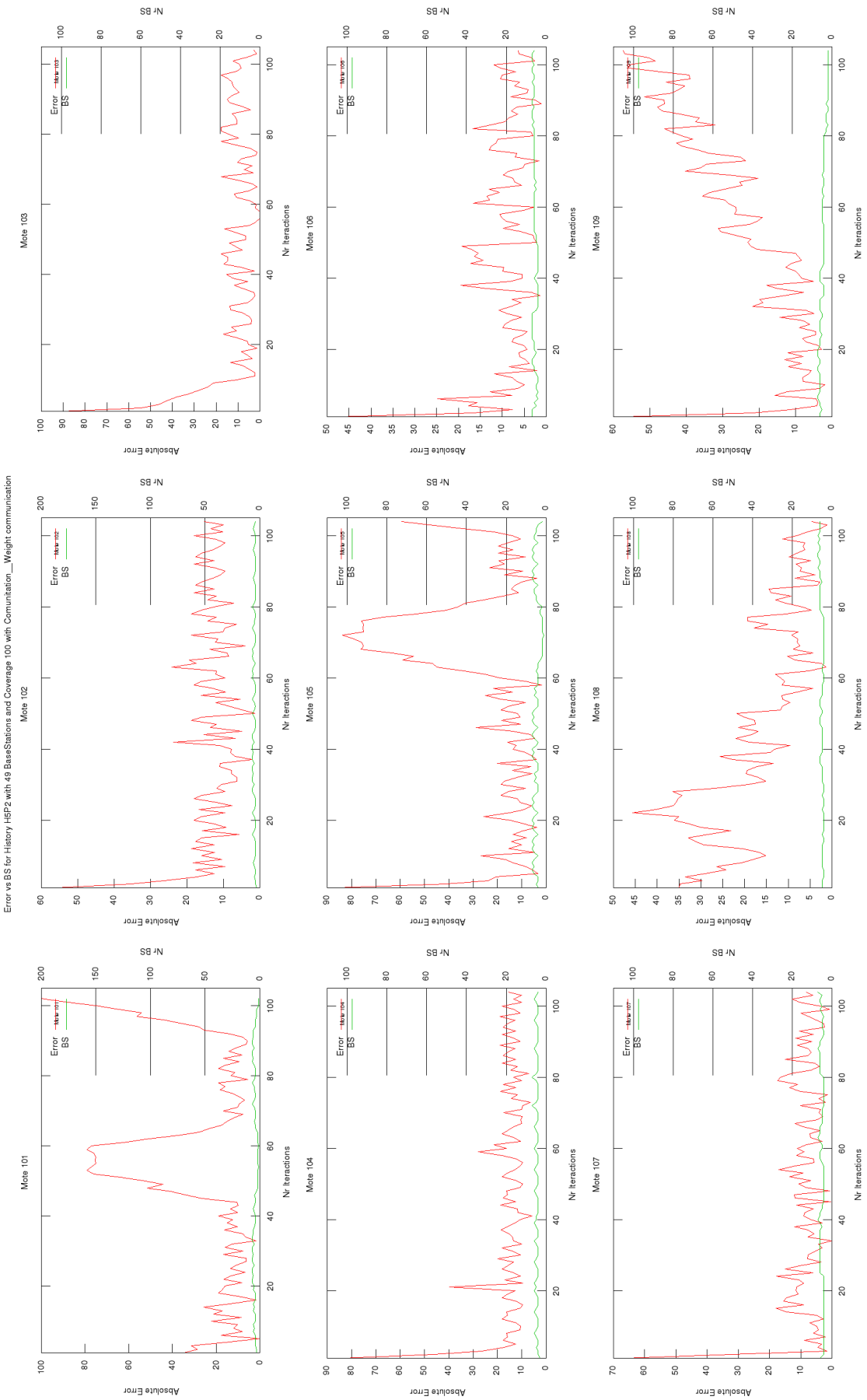


Figure A.20: 49 Base Stations, coverage 100 with previous 5 positions using exponential historical information.

Appendix B

Results in detail, with 100 Base Stations and coverage 100

Here are presented the graphical evolution in detail, on all the cases studied, of the error during the 100 steps for each mote of the 9 motes movements considered, in the case there are 100 Base Stations distributed on the map with coverage 100, since it is with this coverage where the error is minimized.

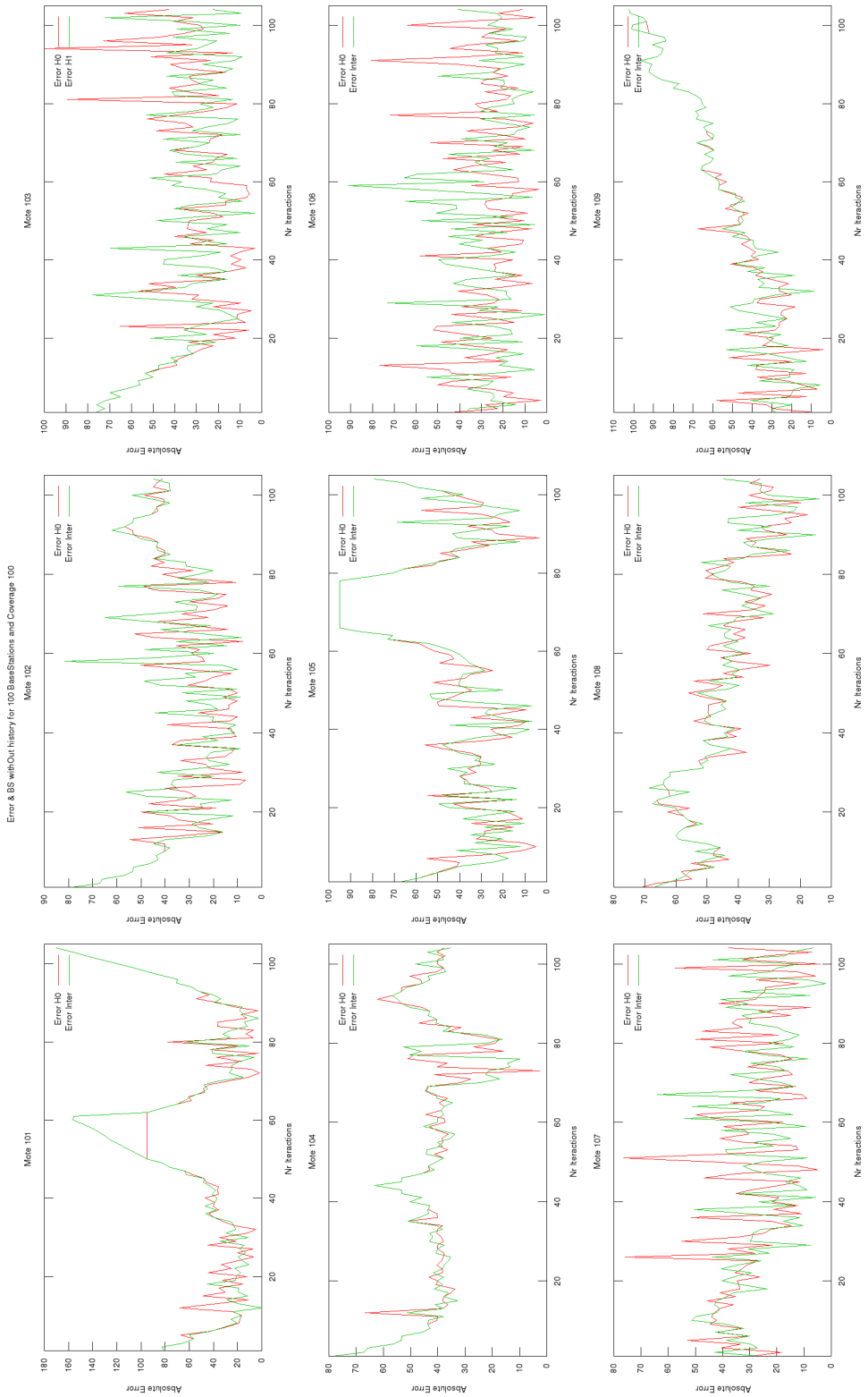


Figure B.1: 100 Base Stations, coverage 100 without historical information.

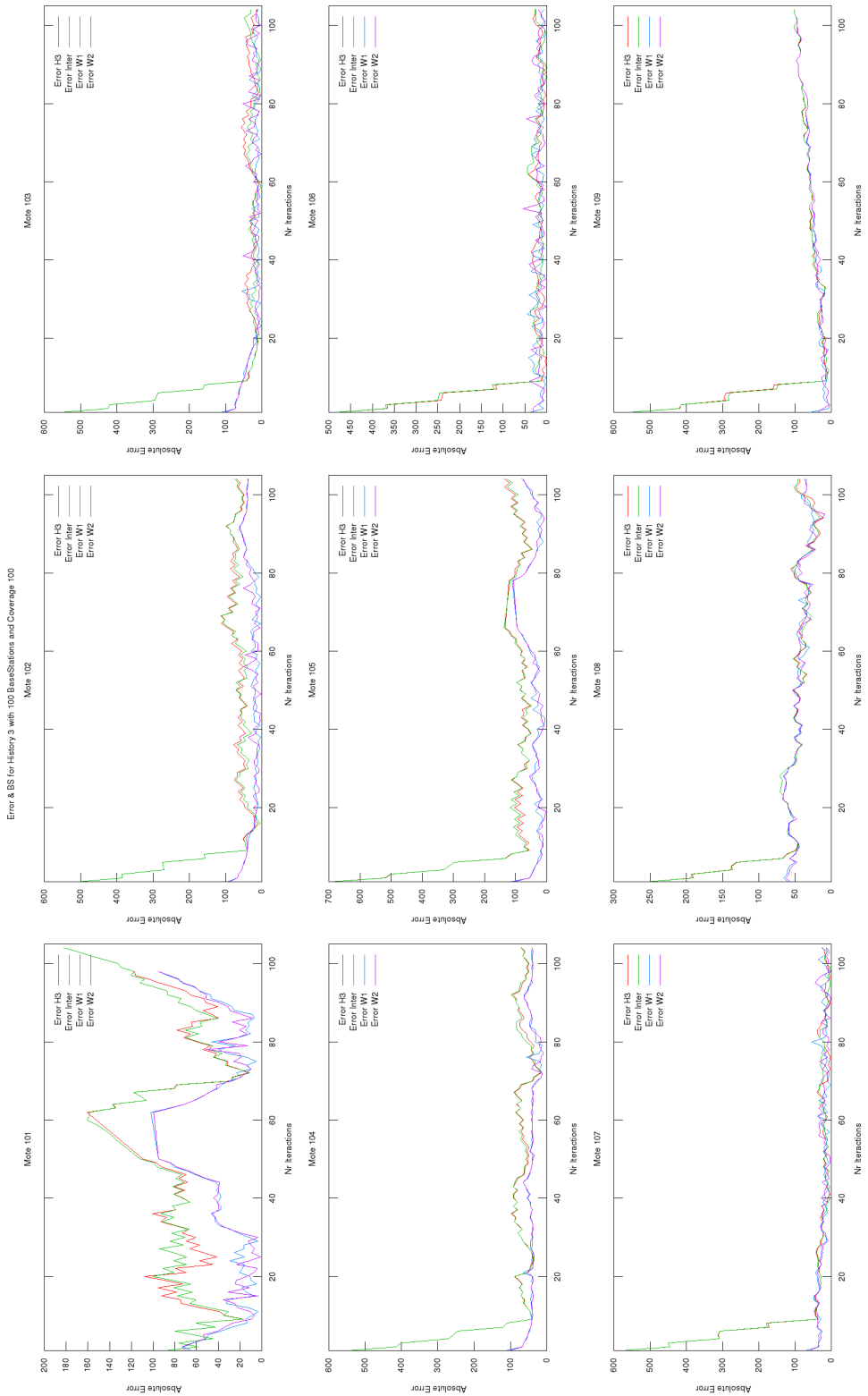


Figure B.2: 100 Base Stations, coverage 100 with 3 previous steps on uniform historical information.

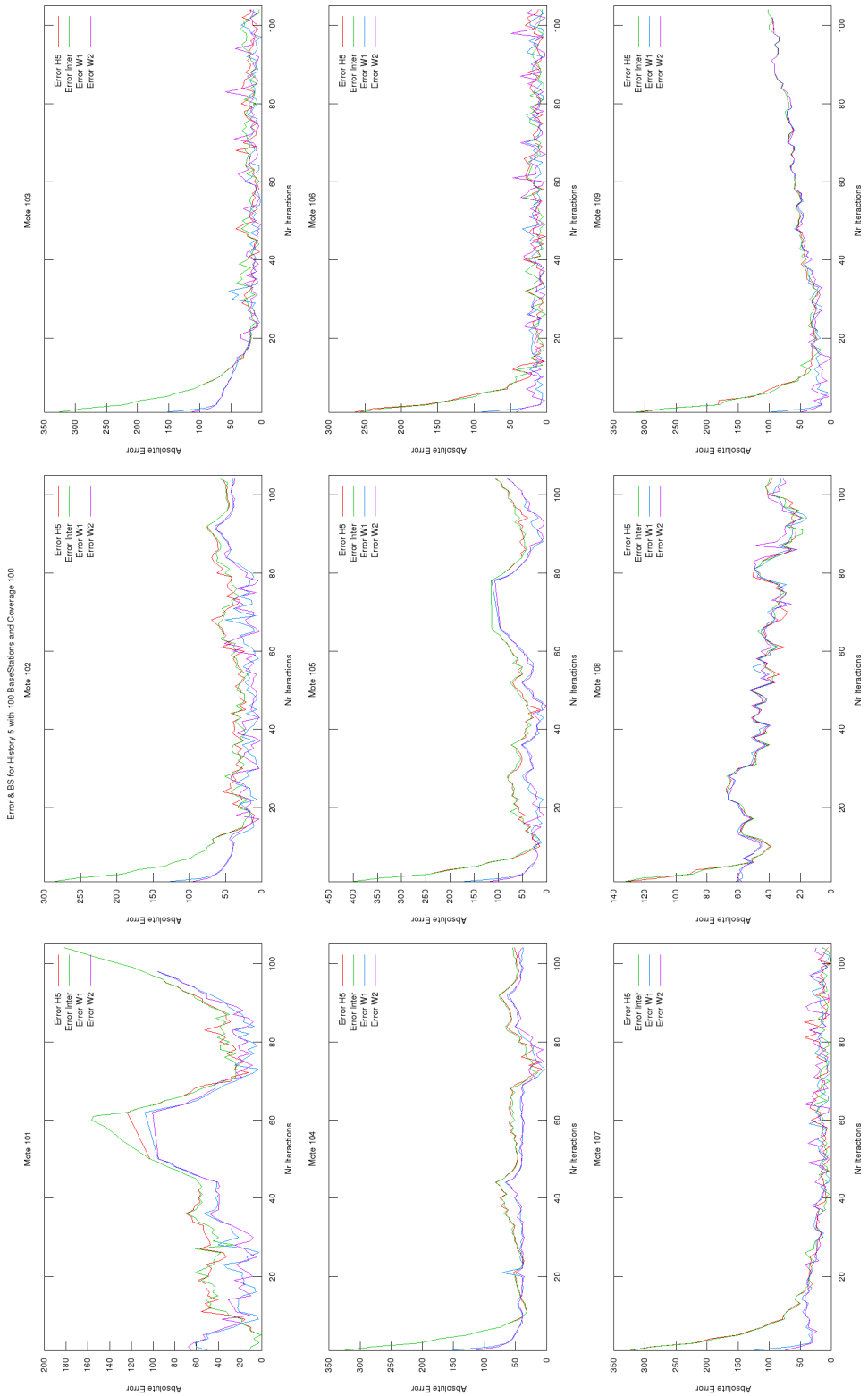


Figure B.3: 100 Base Stations, coverage 100 with 5 previous steps on uniform historical information.

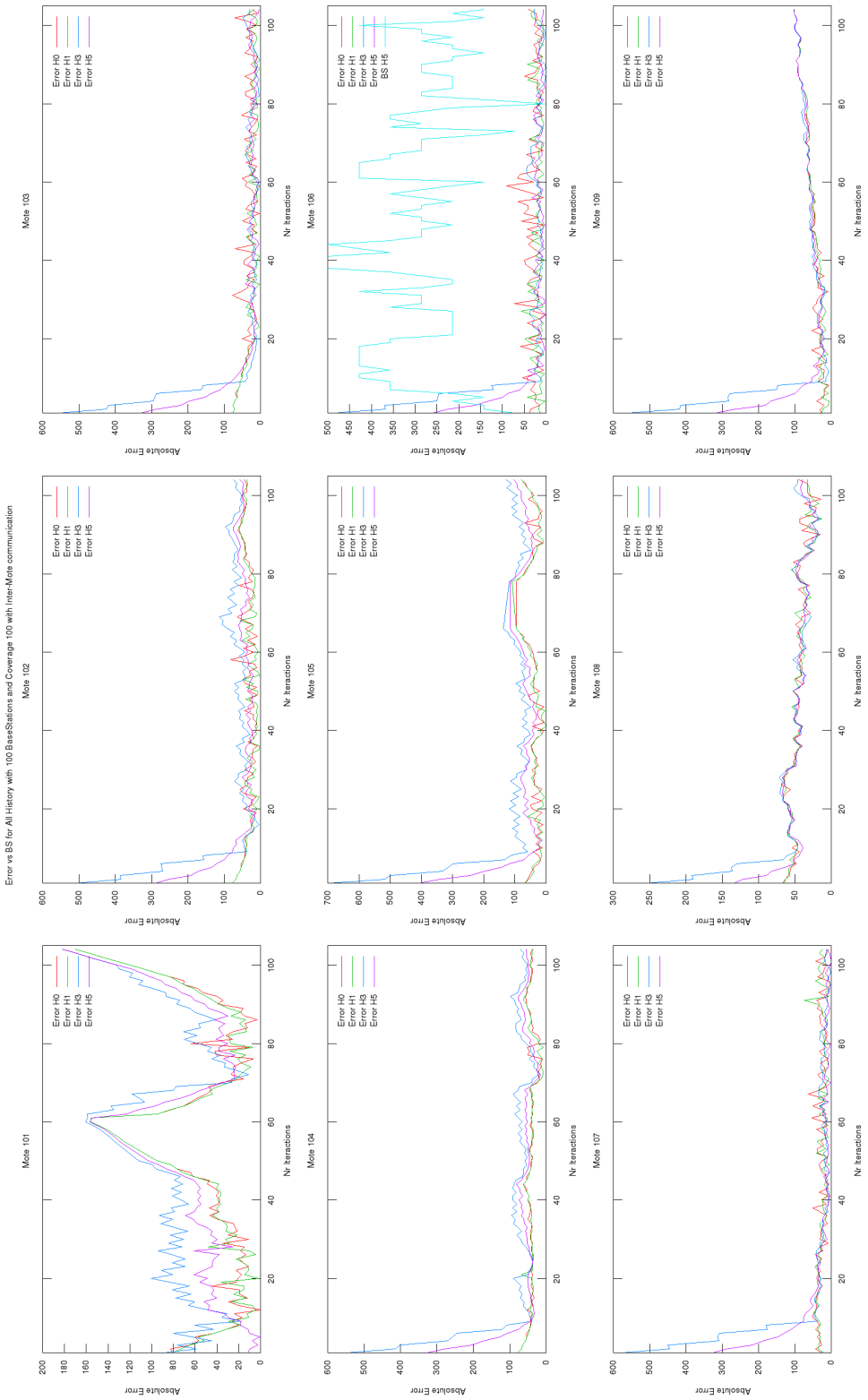


Figure B.4: 100 Base Stations, coverage 100 without historical information, with intercommunication between motes.

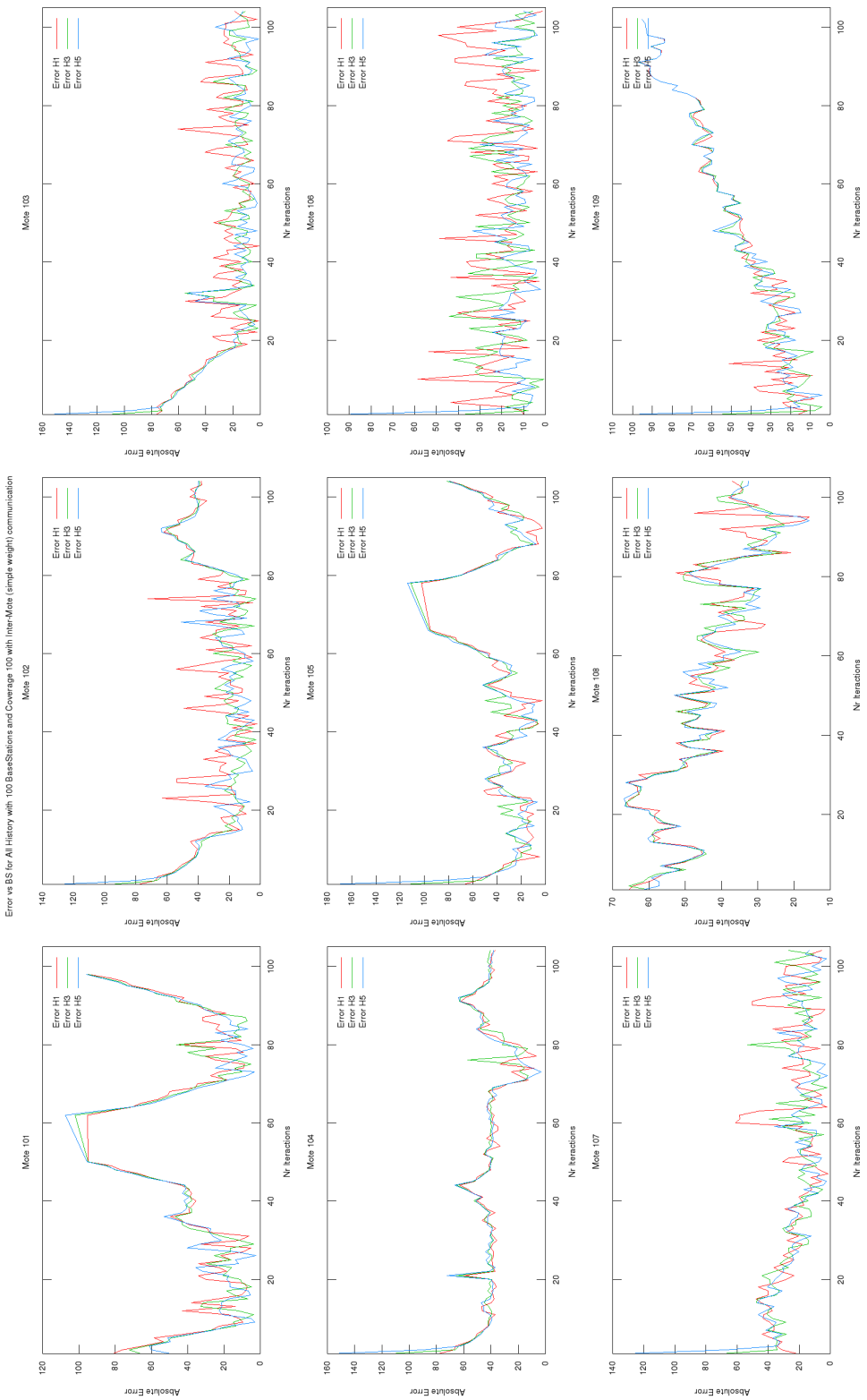


Figure B.5: 100 Base Stations, coverage 100 with 5 previous steps on linear historical information.

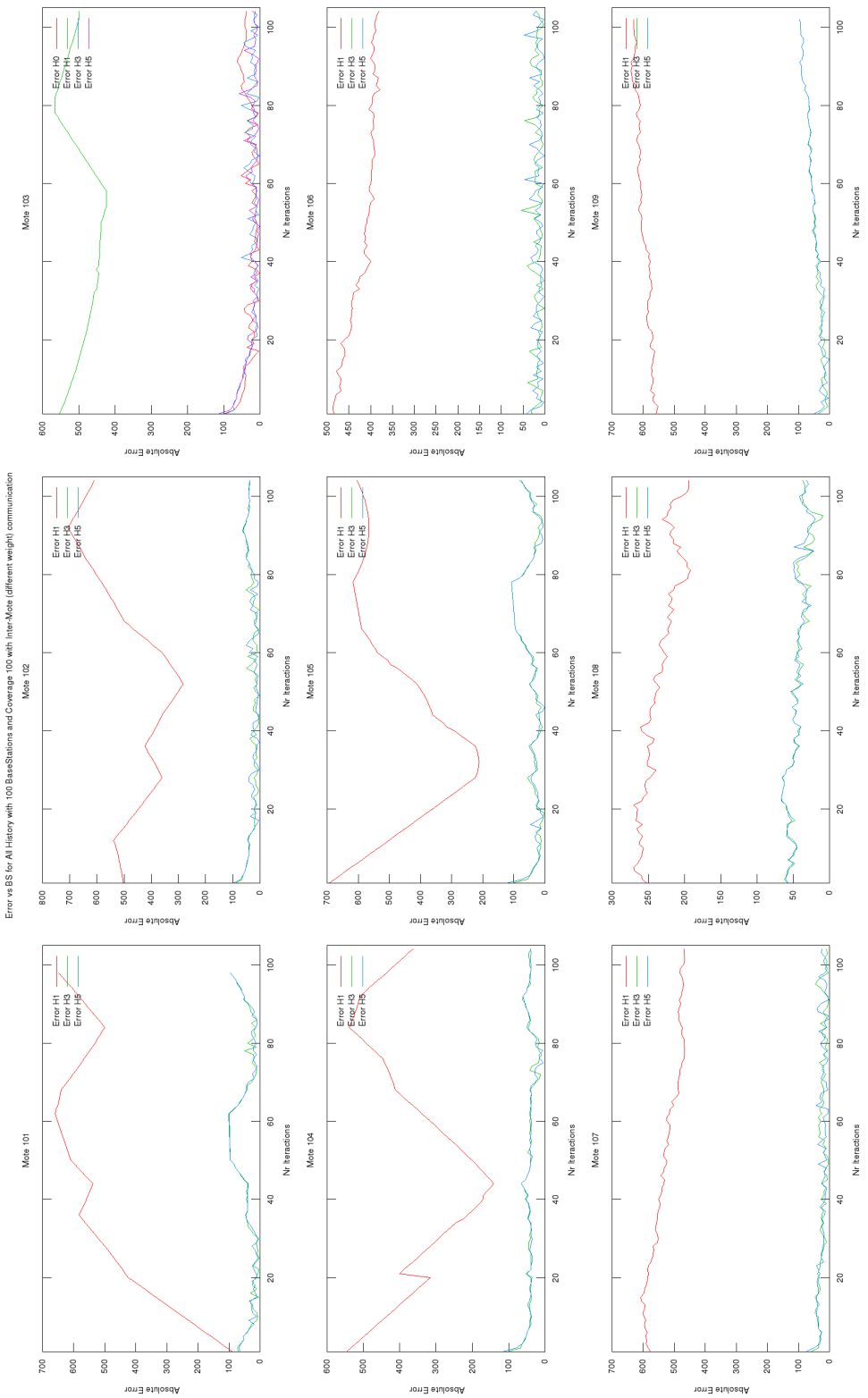


Figure B.6: 100 Base Stations, coverage 100 with 5 previous steps on exponential historical information.

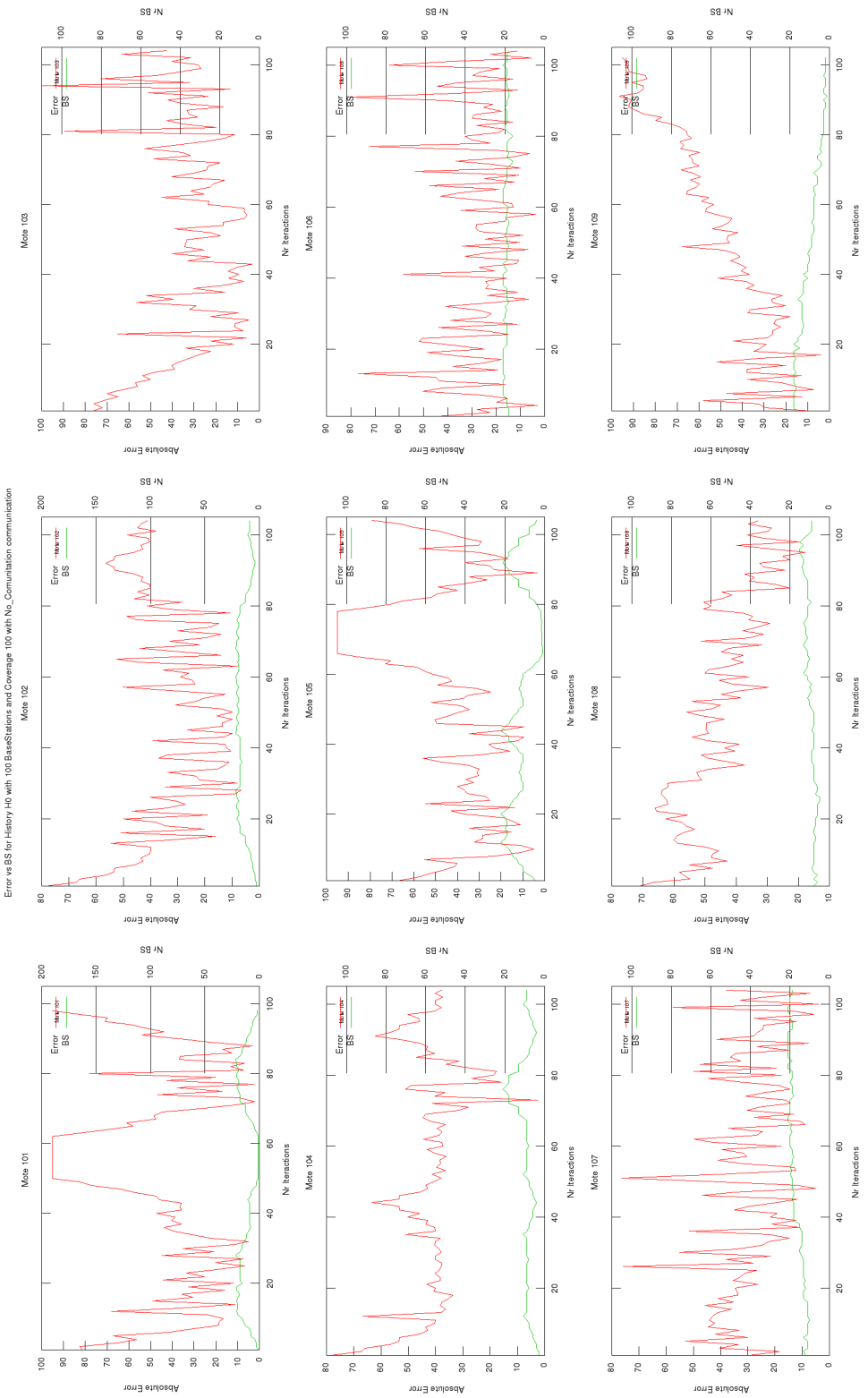


Figure B.7: Pure Centroid with 100 Base Stations and coverage 100.

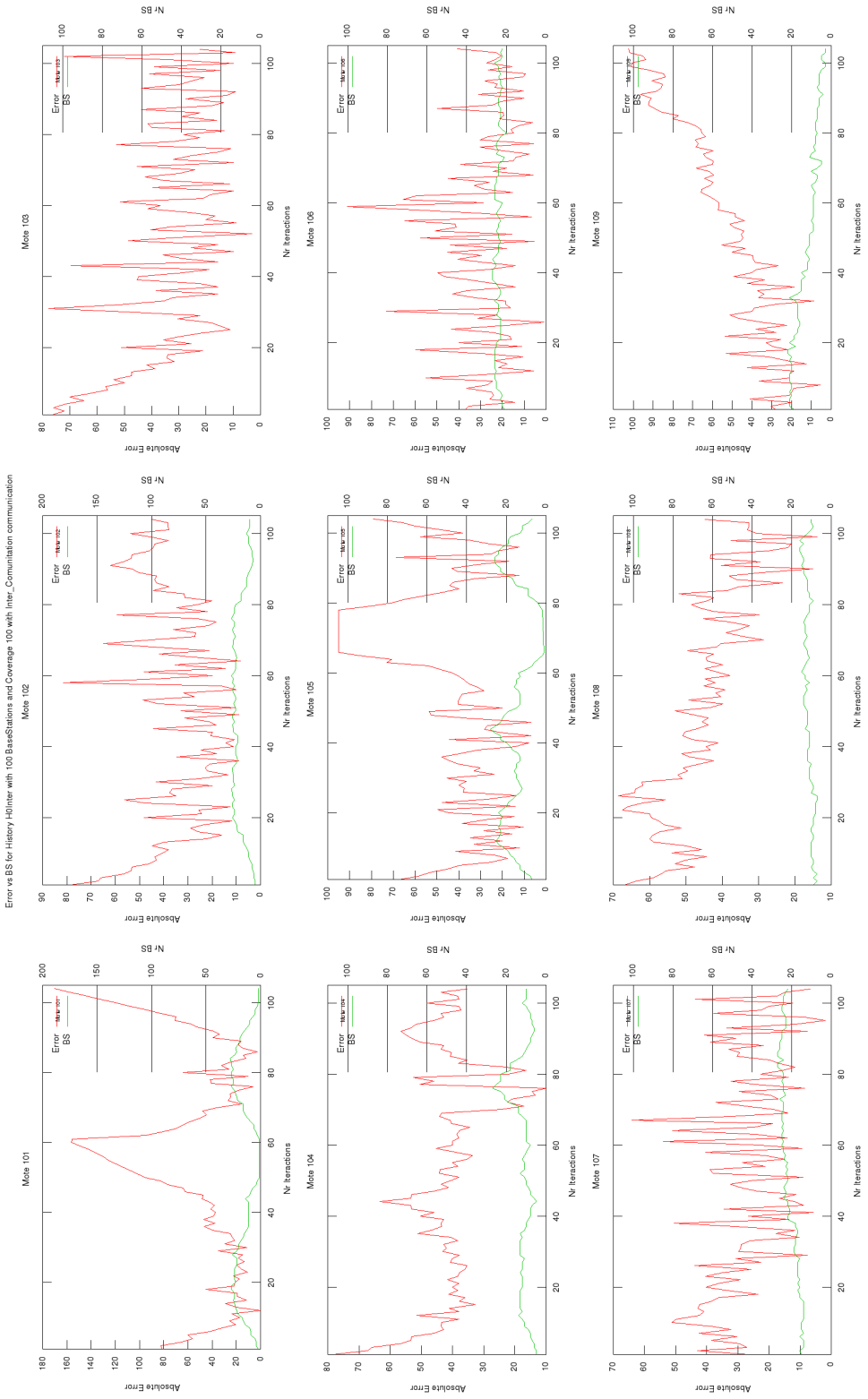


Figure B.8: Pure Centroid with 100 Base Stations and coverage 100, using intercommunication between motes.

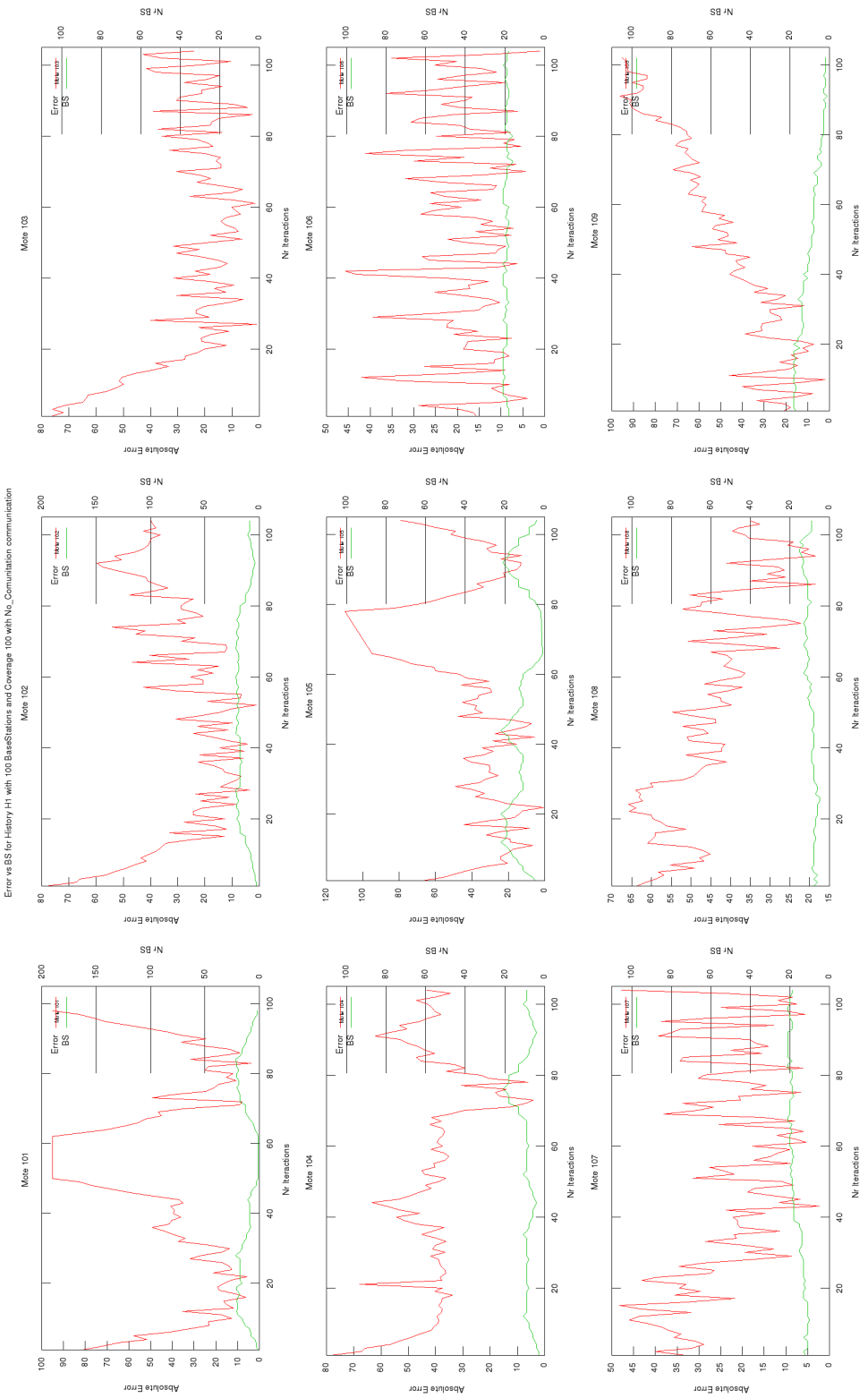


Figure B.9: 100 Base Stations, coverage 100 with last step on uniform historical information.

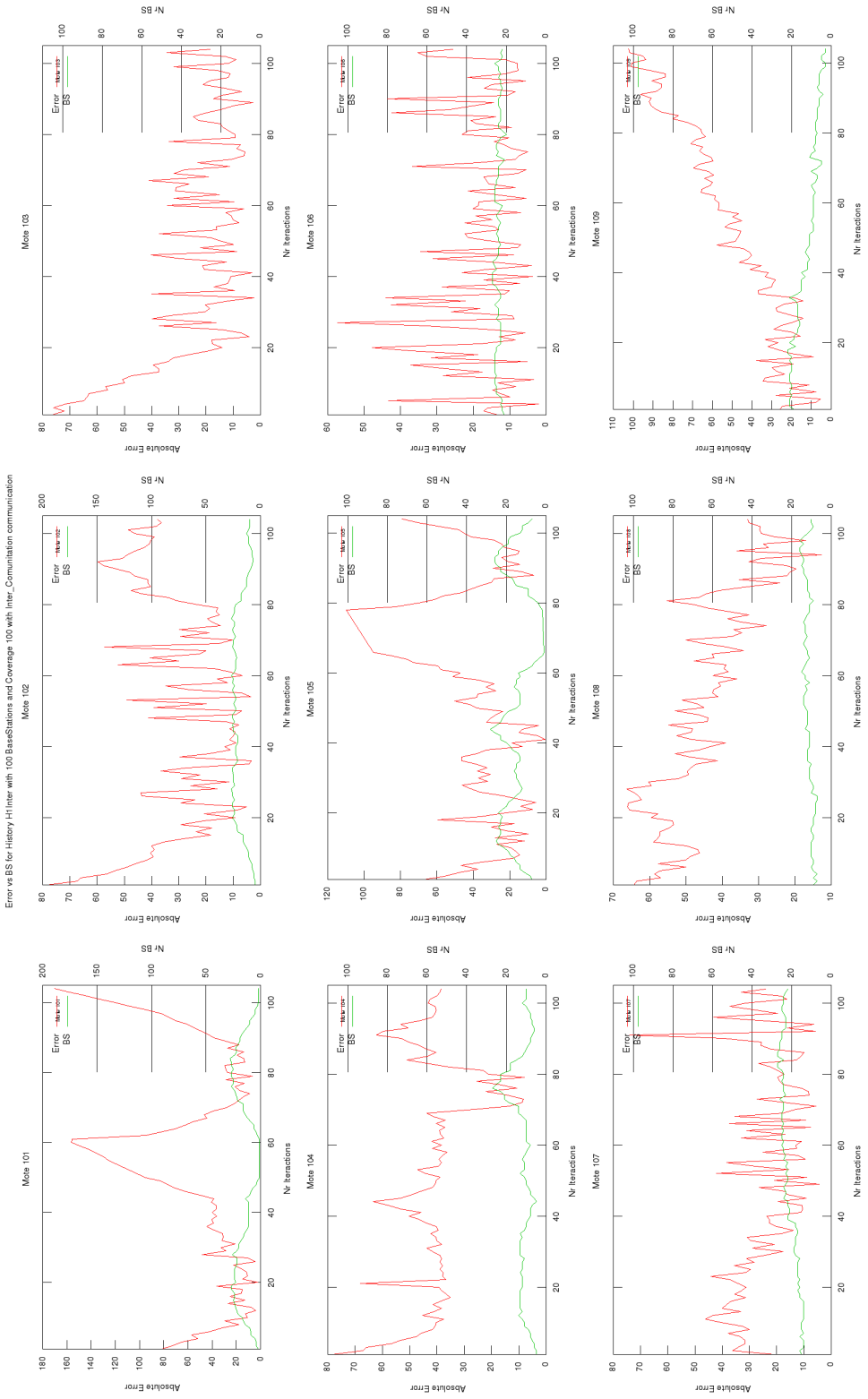


Figure B.10: 100 Base Stations, coverage 100 with last step on uniform historical information, including intercommunication between motes.

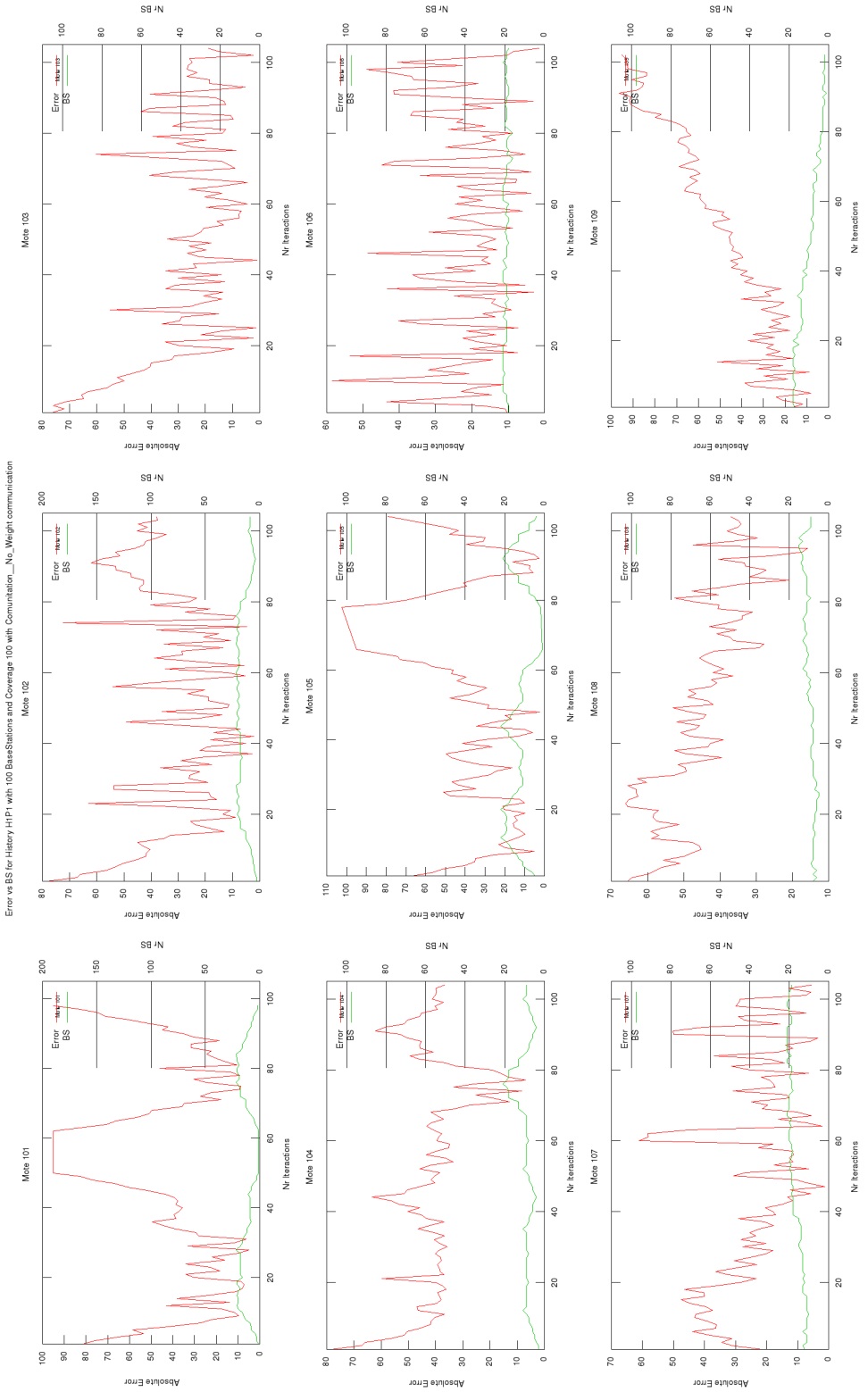


Figure B.11: 100 Base Stations, coverage 100 with last step on linear historical information.

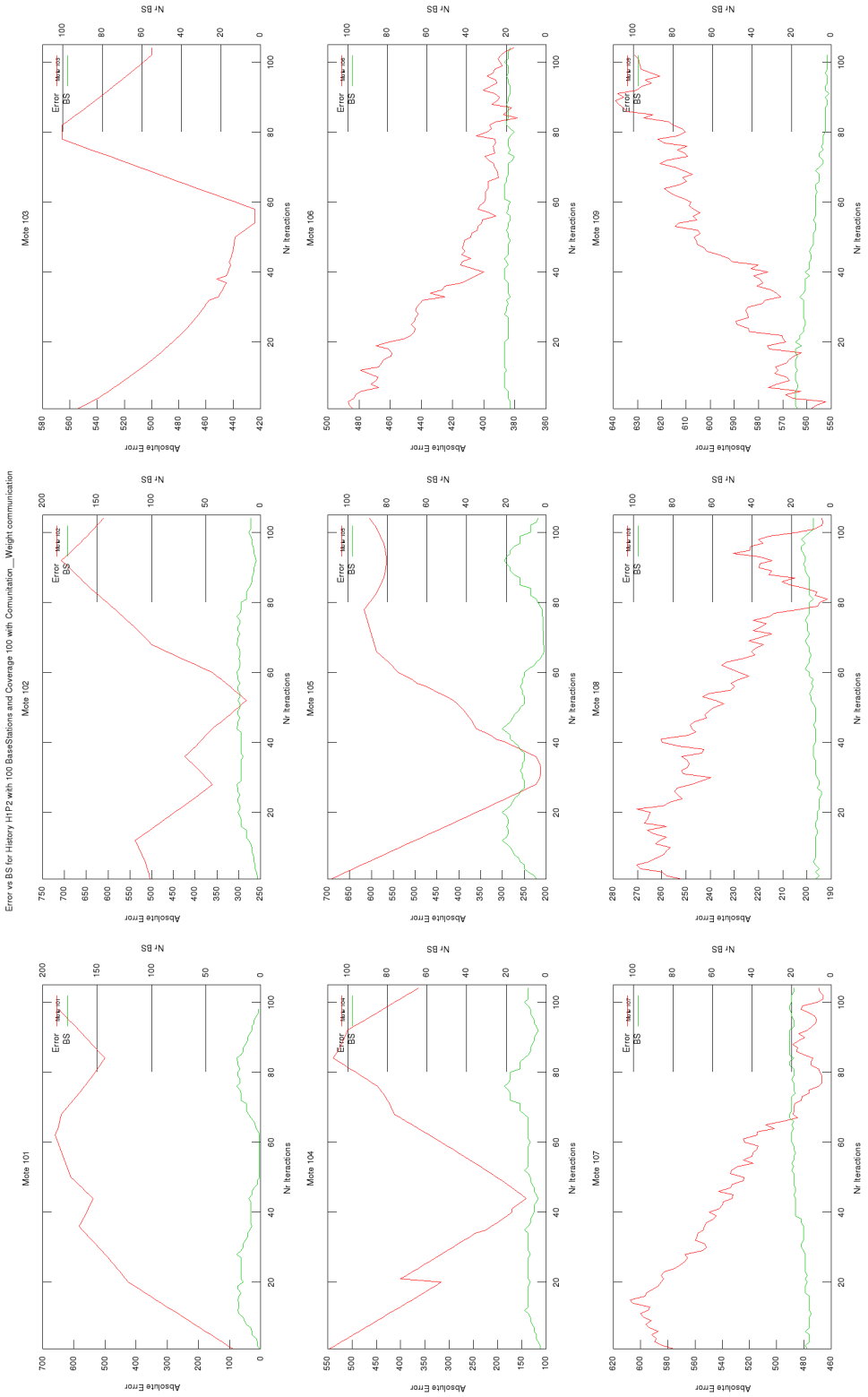


Figure B.12: 49 Base Stations, coverage 100 with last step on exponential historical information.

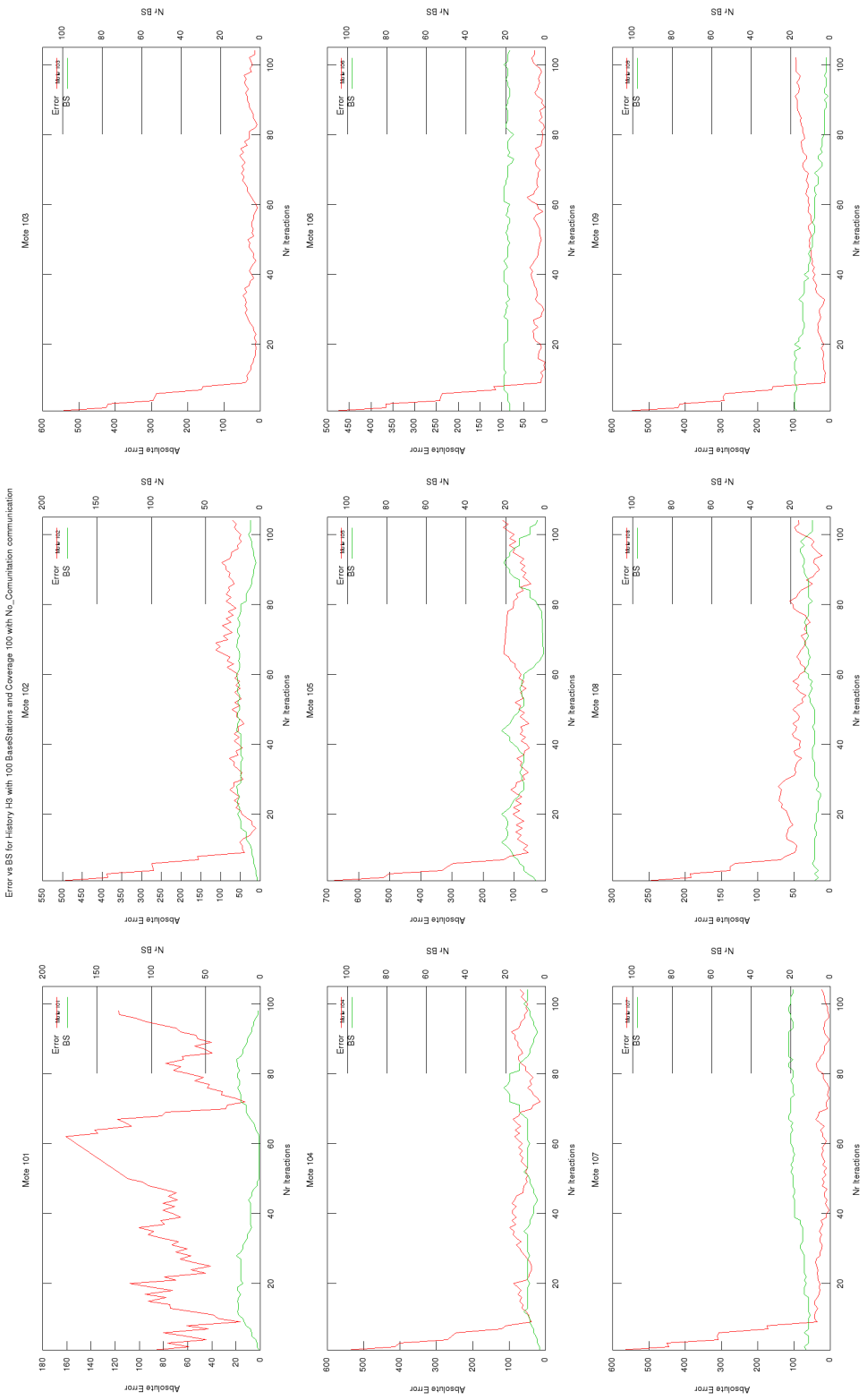


Figure B.13: 100 Base Stations, coverage 100 with previous 3 positions using uniform historical information.

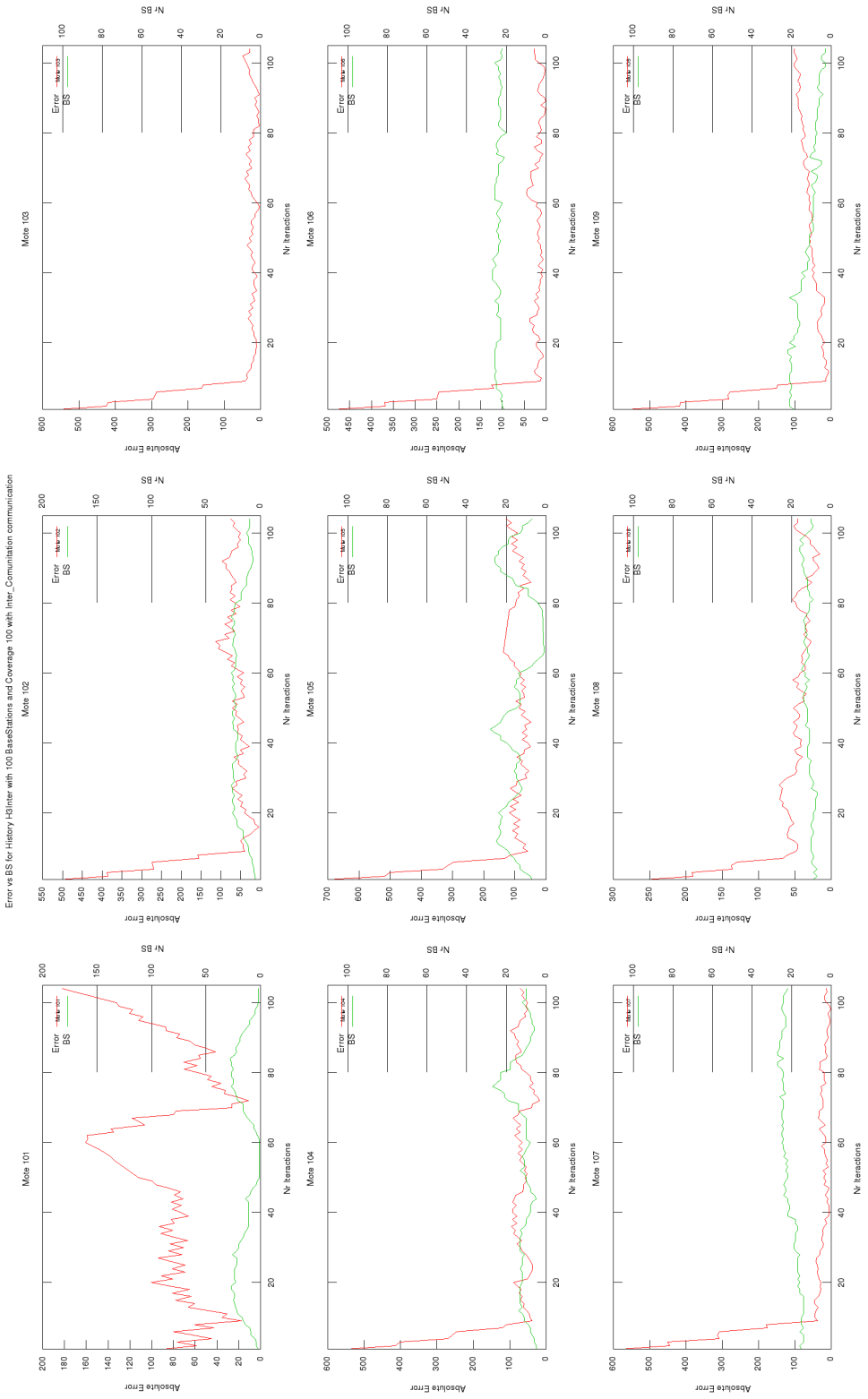


Figure B.14: 100 Base Stations, coverage 100 with previous 3 positions using uniform historical information and intercommunication between motes.

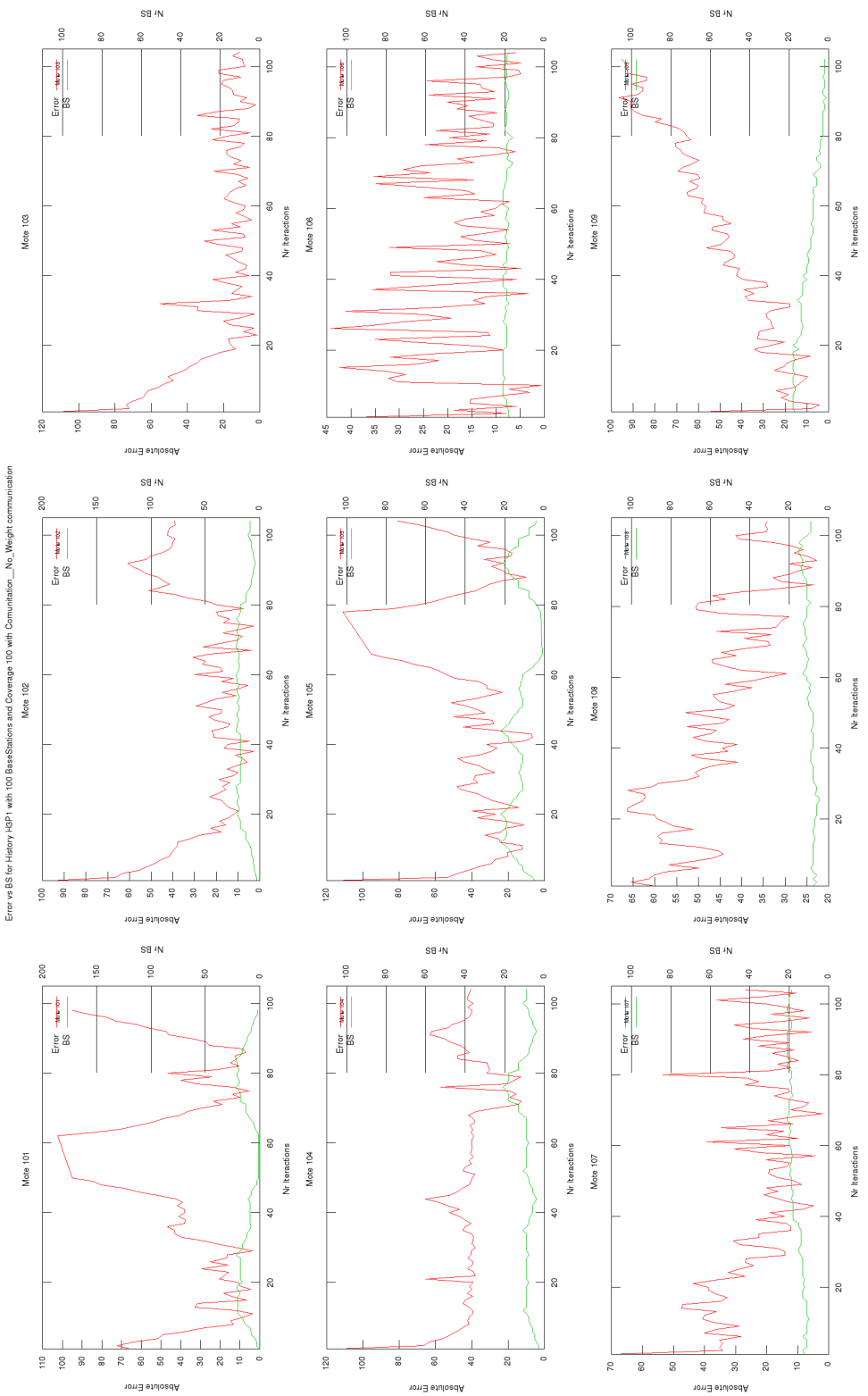


Figure B.15: 100 Base Stations, coverage 100 with previous 3 positions using linear historical information.

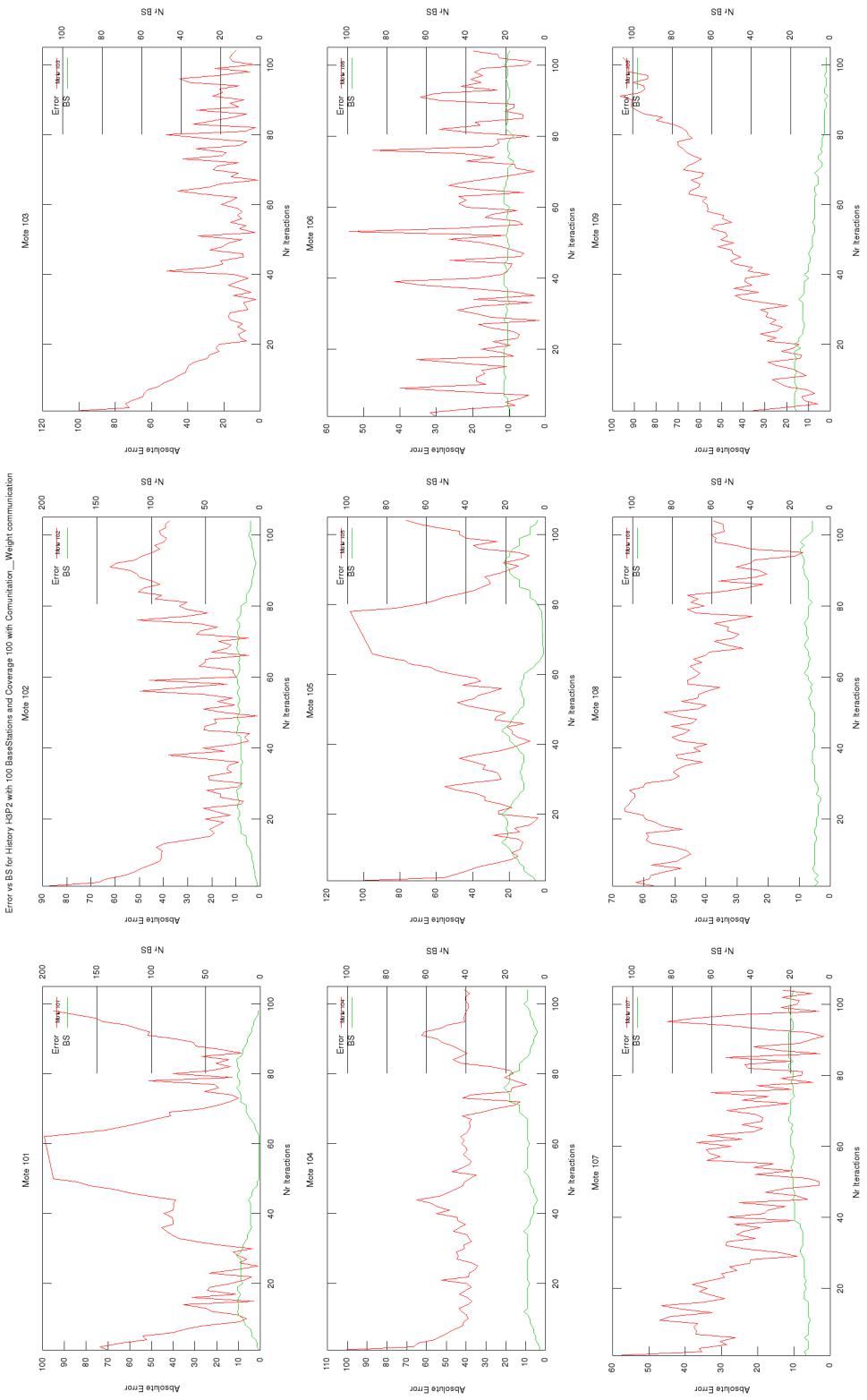


Figure B.16: 100 Base Stations, coverage 100 with previous 3 positions using exponential historical information.

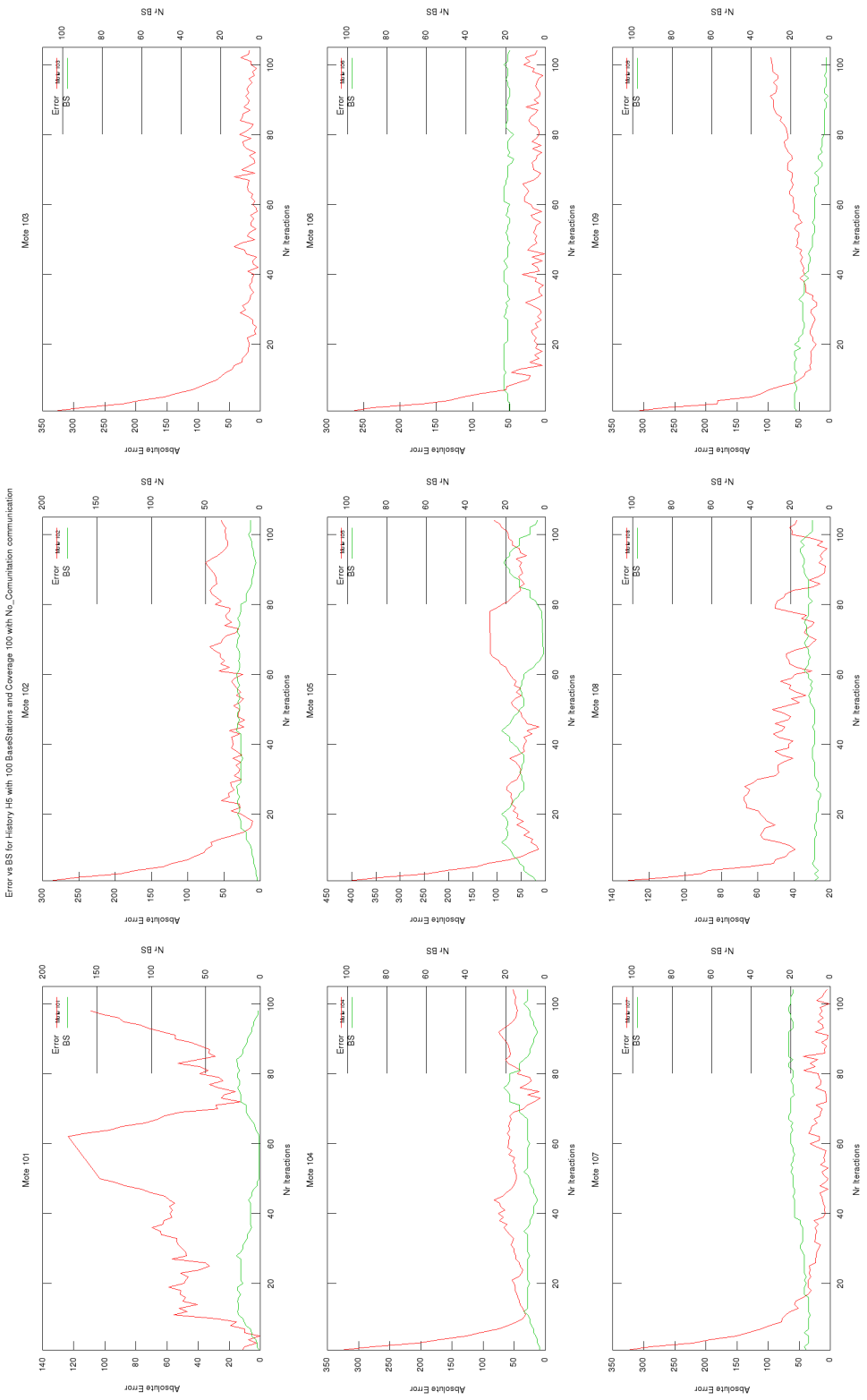


Figure B.17: 100 Base Stations, coverage 100 with previous 5 positions using linear historical information.

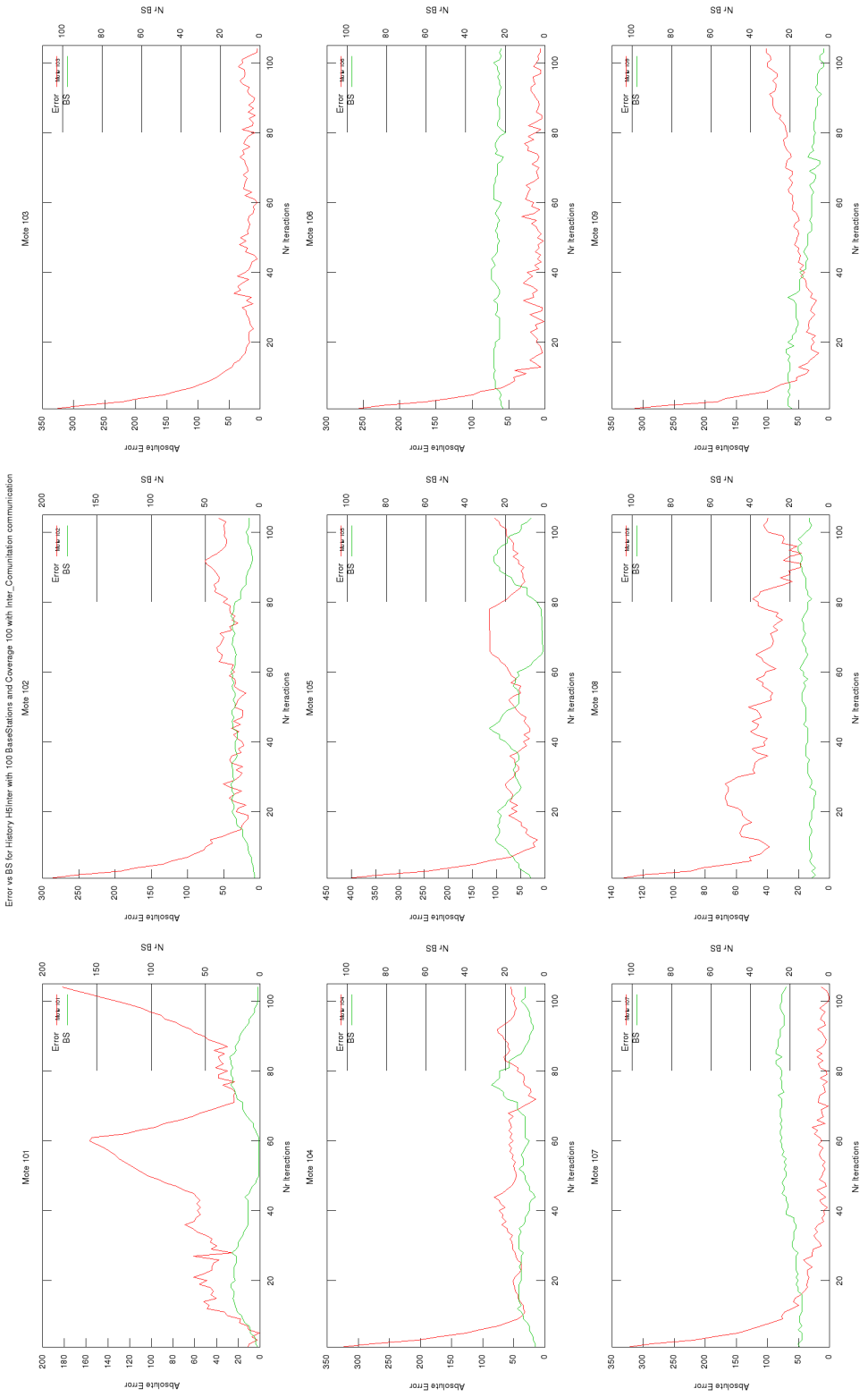


Figure B.18: 100 Base Stations, coverage 100 with previous 5 positions using linear historical information and with intercommunication between motes.

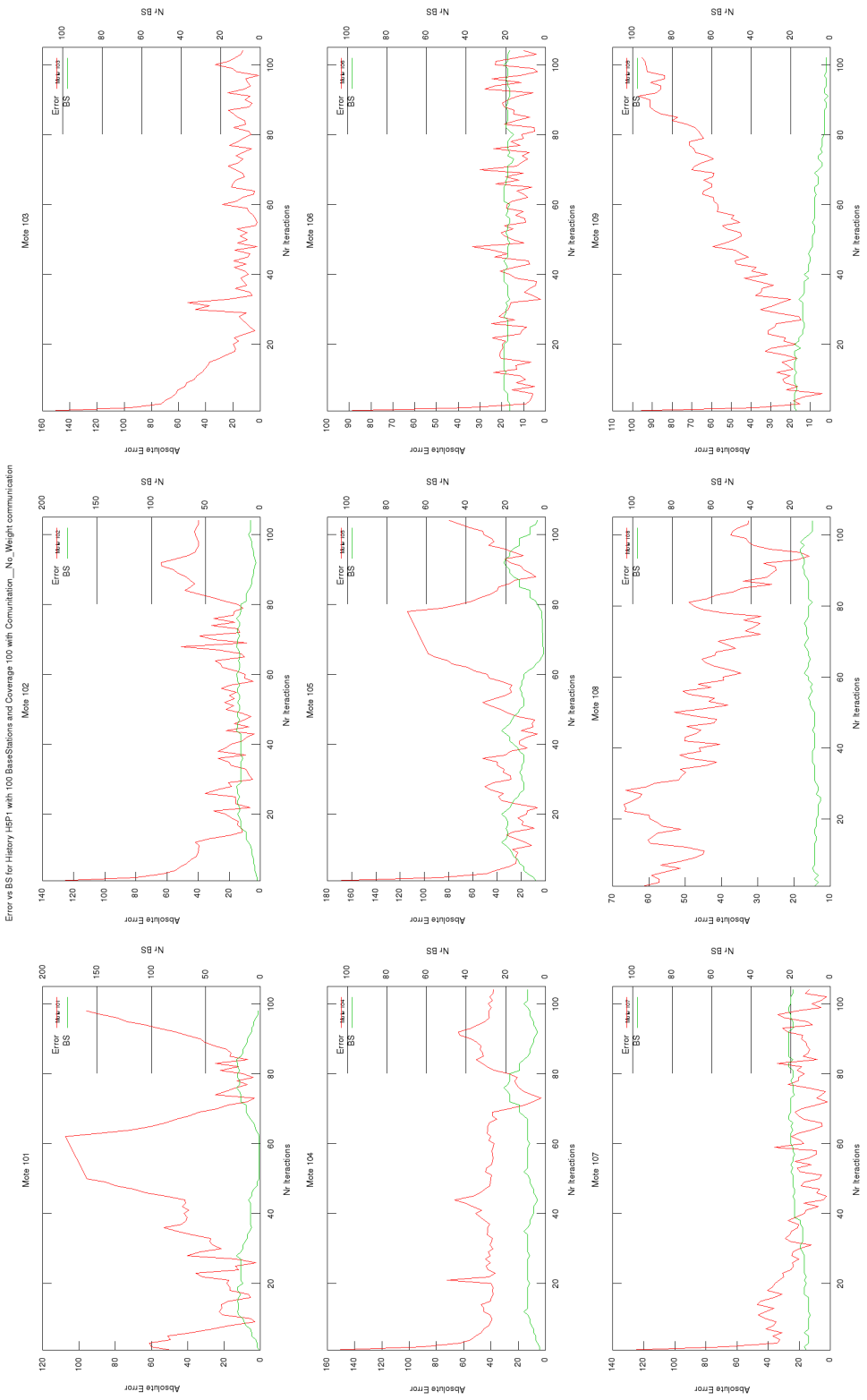


Figure B.19: 100 Base Stations, coverage 100 with previous 5 positions using linear historical information.

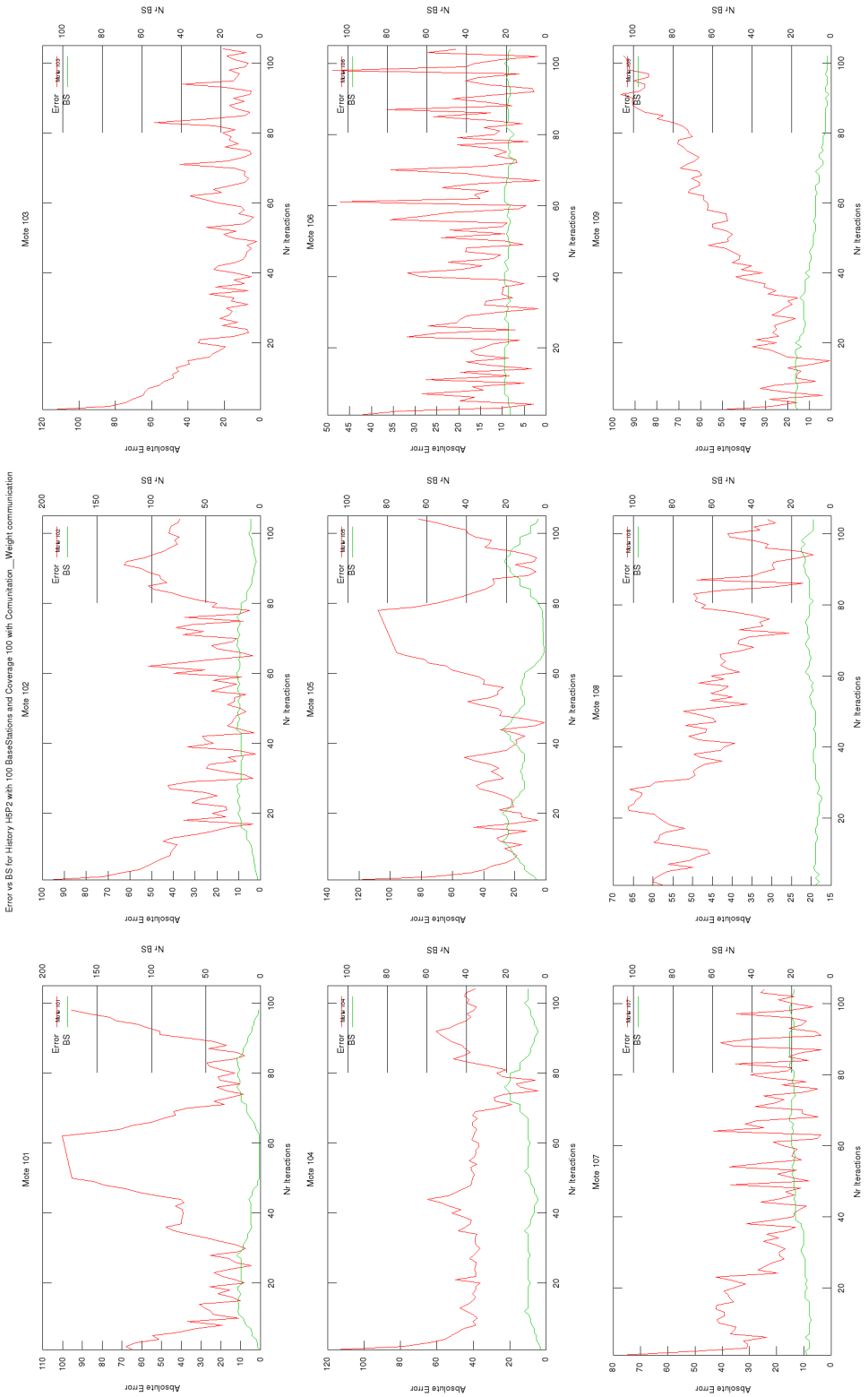


Figure B.20: 100 Base Stations, coverage 100 with previous 5 positions using exponential historical information.