Success Rates of Estimators of Integer Parameters in Box-constrained Linear Models

Sevan Hanssian

Master of Science

School of Computer Science

McGill University

Montreal, Quebec

July 2012

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Science in Computer Science.

DEDICATION

To my mother and father, my brother, my grandmother and grandfather for their unconditional love and undying support, and for believing in me. And to my favourite teacher, Mr. Ghazi.

"No man is an island entire of itself." John Donne

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor, Professor Xiao-Wen Chang, for guiding me as I developed a growing interest in this area of research. I would not have been able to complete my studies without his teachings and advice, his patience and understanding, and his encouragement and support. I am extremely grateful to Jinming for reading my drafts and providing invaluable feedback on and help with my work. I would also like to thank David and Mazen, for welcoming me to the Scientific Computing Lab and for the interesting talks we had; Xiaohu for his help in explaining how to run simulations to test results; Wen-Yang, Stephen, Yancheng, Wanru, Xi, and Shaowei, for their friendship and for the fun times we had; and Milena for becoming one of my closest friends at McGill. Furthermore, I would like to thank Joseph Vybihal for his invaluable advice and support along the way, and my friend Emmanuel in particular for his help with the French abstract. And I would like to express my deepest gratitude to my family for everything they have done and continue to do for me.

Lastly, I wish to thank the students, staff and faculty in the School of Computer Science, and at McGill, for helping to make these the best years of my life so far!

ABSTRACT

In some applied areas such as the Global Positioning System (GPS) and communications, etc., there is a linear model $y = Ax + v$, where the unknown parameter vector x is an integer vector and the noise vector v follows a normal distribution $N(0, \sigma^2 I)$. The typical methods for estimating x are the integer rounding (IR), the Babai nearest plane (BNP), and the integer least squares (ILS) methods. While IR and BNP are polynomial-time methods, the ILS method solves an NP-hard problem. The most effective approach to validating an integer estimator is to find its success rate, which is the probability of correct integer estimation. It has been found in the literature that the ILS estimator is optimal among all admissible integer estimators, including the IR and BNP estimators, as it maximizes the success rate. In communications applications, the integer parameter vector x is often constrained to a box. In this thesis, we first extend the concept of success rates to box-constrained versions of IR (BIR), BNP (BBNP), and ILS (BILS). We then extend some results for the success rates of the corresponding unconstrained estimators to these box-constrained estimators. In addition, we apply the success rate results to improve the efficiency of the BILS estimation process. If some entries of the integer estimator obtained by BBNP have high success rates, then we can fix these entries and solve a smaller BILS problem. This may reduce the overall computational time. Numerical simulations results are presented to support our findings.

ABRÉGÉ

Dans certains domaines appliqués comme la navigation utilisant le système mondial de positionnement (GPS) et comme les communications, il y a un modèle linéaire, $y = Ax+v$, où x est un vecteur de paramètres entiers à estimer et v est un vecteur contenant le bruit et suivant une loi normale $N(0, \sigma^2 I)$. Les méthodes typiques pour estimer x sont l'algorithme d'arrondissement à l'entier le plus proche (IR), l'algorithme de l'hyperplan le plus proche de Babai (BNP) et l'algorithme des moindres carrés en nombres entiers (ILS). Tandis que les algorithmes IR et BNP sont polynomiaux, la méthode ILS résout un problème NP-difficile. L'approche la plus efficace pour valider un estimateur entier est de trouver son taux de réussite, qui est la probabilité de trouver la bonne estimation pour le vecteur de paramètres. Dans la littérature, il s'avère que l'estimateur ILS est optimal parmi les estimateurs admissibles, y compris les estimateurs IR et BNP, car il maximise le taux de réussite. Dans les applications de communication, le vecteur de paramètres est souvent contraint à une boîte. Dans cette thèse, nous développons le concept de taux de réussite pour les méthodes d'estimation avec contraintes de boîtes (les méthodes BIR, BBNP et BILS). Nous présentons aussi quelques résultats pour les taux de réussite de ces estimateurs contraints à boîtes. De plus, nous appliquons ces résultats afin d'améliorer l'efficacité de l'algorithme BILS. Si certaines entrées du vecteur obtenu par BILS ont des taux de réussite élevés, nous pouvons les utiliser et résoudre un problème BILS de dimensions réduites. Cela peut réduire le temps de calcul. Des résultats de simulations sont présentés à l'appui de nos conclusions.

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

$\check{x}$	Integer Estimator
$\check{x}^C$	Box-constrained Integer Estimator
$\hat{x}$	RLS Estimator
$\check{x}^R$	IR Estimator
$\check{x}^B$	BNP Estimator
$\check{x}^I$	ILS Estimator
$\check{x}^{RC}$	BIR Estimator
$\check{x}^{BC}$	BBNP Estimator
$\check{x}^{IC}$	BILS Estimator
P_S	Success Rate
P_S^C	Box-constrained Success Rate
P_{PS}	Partial Success Rate
P_{PS}^C	Box-constrained Partial Success Rate
P_S^R	IR Success Rate
P_S^B	BNP Success Rate
P_S^I	ILS Success Rate
P_S^{RC}	BIR Success Rate
P_S^{BC}	BBNP Success Rate
P_S^{IC}	BILS Success Rate
P_{PS}^B	Partial BNP Success Rate
P_{PS}^{BC}	Partial BBNP Success Rate
$\mathbb{S}$	Pull-in Region
$\mathbb{S}^C$	Box-constrained Pull-in Region

$\mathbb{S}^R$	IR Pull-in Region
$\mathbb{S}^B$	BNP Pull-in Region
$\mathbb{S}^I$	ILS Pull-in Region
$\mathbb{S}^{RC}$	BIR Pull-in Region
$\mathbb{S}^{BC}$	BBNP Pull-in Region
$\mathbb{S}^{IC}$	BILS Pull-in Region
<u>Abbr</u>	<u>Abbreviation</u>
ADOP	Ambiguity Dilution of Precision
BBNP	Box-constrained Babai Nearest Plane
BILS	Box-constrained Integer Least Squares
BIR	Box-constrained Integer Rounding
BNP	Babai Nearest Plane
CVP	Closest Vector Problem
GPS	Global Positioning System
ILS	Integer Least Squares
IR	Integer Rounding
LLL	Lenstra-Lenstra-Lovász
PDF	Probability Density Function
PMF	Probability Mass Function
RLS	Real Least Squares
SQRD	Sorted QR Decomposition
V-BLAST	Vertical Bell Labs Layered Space-Time
VC	Variance Covariance

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

We first describe the notation used in this thesis. We then present the general linear model with unknown integer parameter vector x and give an overview of the typical methods for estimating x and for validating the corresponding estimators. We also detail the contributions of this thesis.

1.1 Notation

The sets of all real and integer $m \times n$ matrices are denoted by $\mathbb{R}^{m\times n}$ and $\mathbb{Z}^{m\times n}$, respectively. The set of real and integer n-vectors are denoted by $\mathbb{R}^n$ and $\mathbb{Z}^n$, respectively. Upper case letters denote matrices and lower case letters denote vectors (or scalars). We use a_i to denote the i^{th} entry of vector a, a_{ij} the $(i,j)^{th}$ entry of matrix $A = [a_{ij}] \in \mathbb{R}^{m\times n}$. We use $D = \text{diag}(d_1, ..., d_n)$ to denote a diagonal matrix. The identity matrix is denoted by I and its i^{th} column is denoted by e_i. The determinant of a matrix A is denoted by $\det(A)$. In addition, $||a||_2 = (a^T a)^{\frac{1}{2}}$ denotes the 2-norm (or Euclidean norm) of vector a, $||a||_\Sigma = (a^T \Sigma^{-1} a)^{\frac{1}{2}}$ denotes the weighted norm of vector a, where Σ is symmetric positive definite. For scalar $\alpha \in \mathbb{R}$, we use $\lfloor \alpha \rceil$ to denote rounding to its nearest integer, $\lfloor \alpha \rfloor$ to denote taking its floor, $\lceil \alpha \rceil$ to denote taking its ceiling, $|\alpha|$ for its absolute value. For a random vector $a \in \mathbb{R}^n$, $a \sim N(0, \sigma^2 I)$ means that a follows a normal (Gaussian) distribution with 0 mean and variance-covariance (VC-) matrix of $\sigma^2 I$, $a \sim \chi^2(n, 0)$ means that a follows a central Chi-squared distribution with n degrees of freedom; the expected value of a is denoted by $E\{a\}$ and its VC-matrix by $\text{cov}\{a\}$. The gamma distribution is denoted by Γ, while the cumulative distribution function of the standard normal distribution $N(0,1)$ is denoted by $\Phi(\alpha) =$

$\int_{-\infty}^{\alpha} \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}t^2\}dt$. P(E) denotes the probability that an event E takes place.

1.2 Unconstrained Problems

In many applications, we have the following linear model:

$$y = Ax + v, \ \ v \sim N(0, \sigma^2 I), \tag{1.1}$$

where $y \in \mathbb{R}^m$ is called the measurement vector (or observed vector), $A \in \mathbb{R}^{m \times n}$ is called the design matrix (or observation or model matrix), $x \in \mathbb{Z}^n$ is called the integer parameter vector, containing unknown integer entries, and $v \in \mathbb{R}^m$ is the measurement noise vector with known variance σ^2. For the purposes of this thesis we will consider deterministic matrix A with full column rank and deterministic unknown vector x.

Our aim is to estimate the integer parameter vector x. This estimation problem arises from many applications, including the Global Positioning System (GPS), radar imaging, communications, cryptography, lattice design, bioinformatics and finance (see, e.g., [1], [11]). Typical estimation methods include integer rounding (which we refer to in this thesis as IR for simplicity), the Babai nearest plane algorithm (which we refer to in this thesis as BNP), and integer least squares (which we refer to as ILS). An "estimator" is random, while an "estimate" is a realization of an estimator. For simplicity, we just use the term "estimator" in this thesis.

Integer rounding (IR), as it is usually called in GPS literature (see, e.g., [11], [22]), involves computing the real least squares (RLS) estimator to x in (1.1) by solving the following RLS problem:

$$\min_{x \in \mathbb{R}^n} ||y - Ax||_2^2. \tag{1.2}$$

Here for simplicity we slightly abuse the notation, as x in (1.2) is a variable while x in (1.1) is an integer parameter vector. By rounding each entry of the RLS solution to the nearest integer, we can obtain an integer estimate to the unknown $x \in \mathbb{Z}^n$ in (1.1). In multi-input multi-output (MIMO) communications, this method is referred to as the linear zero-forcing (ZF) decoder, or the ZF detector (see, e.g., [14], [16]).

The Babai nearest plane (BNP) algorithm given in [2] is a sequential integer rounding process that finds the integer estimate of the i^{th} entry of x, for i from n to 1, by using the previously-obtained integer estimates for the entries from $i+1$ to n. The solution obtained by this method is called the Babai point. In GPS literature, BNP is sometimes referred to as integer bootstrapping [30]. In MIMO communications, the BNP estimator is called the successive interference cancellation (SIC) decoder, which is also known as the decision-feedback detector [14].

The third estimation method is the integer least squares (ILS), where in order to estimate x, we solve the following ILS problem:

$$\min_{x \in \mathbb{Z}^n} ||y - Ax||_2^2. \tag{1.3}$$

In lattice theory, A is referred to as the generator matrix of the lattice $\mathcal{L}(A) = \{Ax : x \in \mathbb{Z}^n\}$. The ILS problem (1.3) is also called the closest (or nearest) vector problem (CVP) or the closest point problem (see, e.g., [1], [16]), since we want to find the point in the lattice closest to the input vector y. In channel coding, the ILS problem is referred to as (sphere) decoding, whereas in source coding, it is called encoding [1].

Unlike the RLS problem, a general ILS problem is NP-hard [3] [15]. This means that all known algorithms for solving (1.3) have exponential complexity. However, as many applications require integer estimates in real-time, we need to be able to solve ILS problems efficiently. A typical approach is the discrete search approach, which involves a reduction phase and a search phase. The main purpose of the reduction phase is to make the search phase more efficient [5]. The Lenstra-Lenstra-Lovász (LLL) reduction strategy which was first presented in [13], but has since been modified in a number of different ways (see [1] and references therein), is very commonly used in practice. LLL reduction may even be used to improve the performance of BNP and thereby find a better Babai point (see, e.g., [14], [16]). As for the search phase, the often used method is the Schnorr and Euchner search strategy [17], which is substantially faster than other available search methods (for a survey of search methods, refer to [1]). This search strategy involves recursively searching the hyper-ellipsoid formed by the ILS problem, in order to find the optimal solution. At each step of the process, the size of the ellipsoidal search space is shrunk using the integer point found in the previous step, and a search for a new integer point is carried out. Eventually, upon failing to find an integer point in the reduced search space, the most recently found one is returned as the optimal solution [5]. The first point generated by the Schnorr and Euchner search is actually the Babai point [1], thus the BNP method is a part of the Schnorr and Euchner search method for solving the ILS problem.

After obtaining an integer estimate of x in (1.1) through one of these methods, we may wish to evaluate its quality or reliability. This process is called validation [20], or verification [11]. In applications like GPS, this validation phase is crucial [20], as explained in the next section. One effective approach to validating the integer estimator involves finding the success rate. The success rate is the probability of correct integer estimation [11] [23], and can be determined once the probability distribution of the corresponding estimator is known [25]. Since it does not depend on the actual outcome of the estimator, validation using success rates is sometimes referred to in the literature as a model-driven approach [26]. If the success rate is higher than a user-defined acceptability threshold, the integer estimator may be trusted and accepted as correct. An important theorem given in [23] states that the ILS estimator is optimal in the class of admissible integer estimators (defined in [23]), which includes the IR and BNP estimators.

Sometimes it may not be possible to successfully resolve all entries of the parameter vector $x \in \mathbb{Z}^n$ [28]. This would require that the n^{th} integer in the estimate obtained coincides with the n^{th} entry of x, that the $(n-1)^{th}$ estimate coincides with the $(n-1)^{th}$ entry, ..., and that the first estimate coincides with the first entry of x. The probability of this simultaneous event tends to decrease as n increases [28]. Hence, if an integer vector estimator has a high success rate, then subsets of its integer entries have high success rates, but if it has a low success rate, this does not necessarily imply that all subsets of its entries have low success rates. The goal of partial validation is to identify the subset that has the highest possible success rate [28]. If this success rate is higher than the user-defined threshold, the entries in the subset are fixed as integers. Estimates to the remaining entries are then obtained by solving updated smaller-sized RLS or ILS problems (depending on the application).

1.3 Example Application: Global Positioning System (GPS)

GPS is a space-based navigation and positioning system that provides one-way ranging from satellites with known positions in space to receivers with unknown positions on land and sea, in air and space [33]. A number of GPS applications require centimeter-level or millimeter-level accuracy in positioning. Examples include monitoring plate motion and crustal deformation for the accurate prediction of earthquakes and volcanic activity (in geodesy or geophysics), or monitoring deformation in large structures such as bridges, towers and dams in real-time (in engineering) [35].

GPS satellites emit signals, which are complex modulated electromagnetic waves containing data, that propagate through space to GPS receivers. The receivers process the signals and measure the time delay, which is the time taken for a signal to propagate from the satellite to the receiver. This is then used to calculate positions. There are two main types of observations in GPS [33]. The code or pseudorange is a coarse measure of the range or distance between the satellite and the receiver antennae. A far more precise observable is the carrier phase measurement. It is equal to the total number of full carrier cycles and fractional cycles between the satellite signal generator at transmission time and the receiver signal correlator at reception time. Although it can accurately measure the fractional phase difference, the GPS receiver cannot distinguish one full carrier cycle from the next, the reason being that carriers are pure sinusoidal waves and each cycle in a sinusoid looks just like the next. The GPS integer ambiguity refers to this unknown integer number of full cycles. Both of these measurements are biased by several factors including atmospheric effects, clock errors and instrumental delays [33].

Any linearized GPS observation model can be written as [20]:

$$y = Aa + Bb + v, \tag{1.4}$$

where $y \in \mathbb{R}^m$ is the observation vector, $a \in \mathbb{Z}^n$ is the vector of unknown carrier phase ambiguities, $b \in \mathbb{R}^p$ is the vector of unknown position coordinates, $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{m \times p}$ are the corresponding design matrices for a and b, and $v \in \mathbb{R}^m$ is the noise vector.

In order to obtain high precision position estimates of centimeter-level accuracy for parameter vector b, we must first estimate integer values for the carrier phase ambiguities of parameter vector a. This is typically achieved by solving an ILS problem [20]. Since most GPS applications require real-time fast positioning, we have to solve the ILS problem efficiently to estimate a. This becomes even more challenging when GPS signals are combined with signals from other satellite systems (like Galileo), thus resulting in larger ILS problems.

A common GPS parameter estimation process found in the literature consists of three steps [30]. First, we compute the RLS estimators $\hat{a}$ and $\hat{b}$ of a and b, respectively. In the second step, we compute the integer estimator $\check{a}$ using IR, BNP or ILS, and finally, we correct the real estimate $\hat{b}$ using the integer vector $\check{a}$, to obtain $\check{b}$. Therefore the position estimates are dependent on the integer ambiguity estimates [20]. If the integers found are not correct, the position estimates will have large errors, and this may be dangerous for certain applications like aircraft landing. Thus, GPS ambiguity resolution consists of two subproblems: ambiguity estimation to compute $\check{a}$, and ambiguity validation to evaluate its quality by finding the success rate. If this is higher than the user-defined threshold, we may decide to accept $\check{a}$ and use it to improve the precision of the position coordinates in b. Otherwise, we reject $\check{a}$ and use the RLS estimator $\hat{a}$ instead in any further computations, as it would be less harmful than using an incorrect integer estimator [26].

1.4 Box-constrained Problems

In some applications such as wireless communications (see, e.g., [9]), x in the linear model (1.1) is subject to the box constraint:

$$\mathbb{B} = \{x \in \mathbb{Z}^n |\ l \leq x \leq u,\ l \in \mathbb{Z}^n, u \in \mathbb{Z}^n\}. \tag{1.5}$$

In order to estimate the box-constrained integer parameter vector x, we solve the following box-constrained integer least squares (BILS) problem:

$$\min_{x \in \mathbb{B}} ||y - Ax||_2^2. \tag{1.6}$$

A typical method for solving (1.6) also involves a reduction (or pre-processing) phase and a search phase. The key part of a reduction algorithm is to reorder the columns of A, as different column permutations may have significantly different effects on the search speed [5]. The vertical Bell Labs layered space-time (V-BLAST) optical detection ordering given in [10] was proposed as a reduction strategy in [9]. The sorted QR decomposition (SQRD) is another reduction strategy proposed in [32] for decoding the same codes as those decoded by V-BLAST algorithms. To solve the BILS problem, the box-constraints must be considered during the search. There is more than one way to modify the unconstrained search strategy to take the box-constraints into account (see, e.g., [4], [5], [9]).

1.5 Thesis Contributions

In this research, we extend the theory of success rates to box-constrained versions of the IR (BIR), BNP (BBNP) and ILS (BILS) estimators. We then apply the extended success rate results to improve the efficiency of the BILS estimation process. More specifically, if some entries of the integer estimate obtained by BBNP have high success rates, then we can fix these entries and solve a smaller-sized BILS problem to obtain better estimates for the remaining entries. Since BBNP is a polynomial-time estimation method while BILS is NP-hard, this partial fixing approach may reduce the overall computational time. We consider the BIR estimator because it is a simple estimator.

This thesis is organized as follows. In chapter 2, we review the three integer estimation methods, IR, BNP and ILS, which are typically used in practice to estimate x in (1.1). We present the pull-in regions, also called Voronoi cells (see, e.g., [11], [34]), of the corresponding estimators. A pull-in region is a subset of $\mathbb{R}^n$ which contains all the real vectors that are mapped, differently depending on the estimator used, to a particular integer vector [23]. In chapter

3, we review the success rates of the IR, BNP and ILS estimators, which are found using their respective pull-in regions and the parameter probability distribution given in [25], and we present some success rate results found in the literature, including the optimality property of the ILS estimator [23]. We also discuss partial success rates. In chapter 4, we modify the integer estimation methods to take the box constraints in $\mathbb{B}$ (1.5) into account when estimating x in (1.1), to obtain box-constrained IR (BIR), BNP (BBNP) and ILS (BILS) estimators. We also extend the concept of pull-in regions to the BIR, BBNP and BILS estimators. In chapter 5, we extend the theory of success rates to box-constrained estimators and consider the properties of the success rates of the BIR, BBNP and BILS estimators. In particular, we give examples to show that some properties, such as the optimality of the ILS estimator, which hold in unconstrained problems, do not hold in box-constrained problems. Furthermore, we apply extended partial success rate results in an attempt to improve the efficiency of the BILS estimation process, thereby reducing overall computational time. Numerical simulations results are presented to support our findings. Finally, in chapter 6, we give some conclusions and discuss possible future work.

CHAPTER 2
Integer Parameter Estimation

We review the IR, BNP and ILS methods to estimate the integer parameter vector in the linear model (1.1), and introduce the concept of pull-in regions.

2.1 Reduced Integer Least Squares Problem

The discrete search approach to solving an ILS problem involves two phases: a reduction and a search, with the purpose of the reduction being to make the search process more efficient. We can transform the original ILS problem (1.3), $\min\limits_{x \in \mathbb{Z}^n} ||y - Ax||_2^2$ where A has full column rank, into a new ILS problem which can be solved more efficiently, by transforming matrix A into an upper triangular matrix R which has some additional properties. This is achieved through the QRZ factorization of A:

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix} Z^{-1} = [Q_1, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} Z^{-1} = Q_1 R Z^{-1}, \qquad (2.1)$$

where $Q = [Q_1, Q_2] \in \mathbb{R}^{m \times m}$ is orthogonal, $R \in \mathbb{R}^{n \times n}$ is nonsingular upper triangular, and $Z \in \mathbb{Z}^{n \times n}$ is unimodular, i.e. $|\det(Z)| = 1$, implying Z^{-1} is also an integer matrix [8]. Then we have

$$\begin{aligned} ||y - Ax||_2^2 = ||Q^T y - Q^T Ax||_2^2 &= \left|\left| \begin{bmatrix} Q_1^T y \\ Q_2^T y \end{bmatrix} - \begin{bmatrix} RZ^{-1}x \\ 0 \end{bmatrix} \right|\right|_2^2 \\ &= ||Q_1^T y - RZ^{-1}x||_2^2 + ||Q_2^T y||_2^2. \end{aligned}$$

Define

$$\bar{y} \triangleq Q_1^T y \in \mathbb{R}^n, \quad \bar{x} \triangleq Z^{-1}x \in \mathbb{Z}^n.$$

We obtain the following reduced ILS problem equivalent to (1.3):

$$\min_{\bar{x}\in\mathbb{Z}^n} ||\bar{y} - R\bar{x}||_2^2. \tag{2.2}$$

After the optimal solution $\check{\bar{x}}$ to (2.2) is found by the Schnorr and Euchner search (see §2.4), we obtain the optimal solution $\check{x}$ to (1.3) by $\check{x} = Z\check{\bar{x}}$.

The process of transforming A to R is called a reduction. We initially compute the QR factorization of A using Householder transformations, Givens rotations or Gram-Schmidt orthogonalization, which makes Z equal to the identity matrix I in (2.1). Then, we apply LLL reduction (see §2.4), updating R by orthogonal transformations from the left, to keep the solution of (1.3) unchanged, and unimodular transformations from the right, to keep the integer nature of the unknown parameter vector x, to eventually obtain the desired reduced upper triangular matrix R [8].

For simplicity of notation, we will use x rather than $\bar{x}$ in (2.2), i.e.,

$$\min_{x\in\mathbb{Z}^n} ||\bar{y} - Rx||_2^2. \tag{2.3}$$

It is therefore important to note that as a final step, the integer estimator obtained by IR, BNP or ILS in the following sections must be left-multiplied by Z to obtain the correct estimator to x in (1.3). We denote the RLS estimator by $\hat{x}$ and an integer estimator by $\check{x}$. Furthermore, we denote the IR estimator by $\check{x}^R$, the BNP estimator by $\check{x}^B$, and the ILS estimator by $\check{x}^I$.

2.2 Integer Rounding (IR) Estimation

The simplest integer estimator for x is obtained by the IR method, which involves rounding the individual entries of the RLS solution of $\min\limits_{x\in\mathbb{R}^n} ||\bar{y} - Rx||_2^2$ to the nearest integers. The RLS estimator $\hat{x}$ satisfies $\bar{y} = R\hat{x}$, and this upper triangular system can be solved by back substitution, starting from the n^{th} equation to obtain $\hat{x}_n = \dfrac{\bar{y}_n}{r_{nn}}$. We then choose $\check{x}_n^R = \lfloor \hat{x}_n \rceil$. Using the $(n-1)^{th}$

equation, we find $\hat{x}_{n-1} = \dfrac{\bar{y}_{n-1} - r_{n-1,n}\hat{x}_n}{r_{n-1,n-1}}$, and $\check{x}^R_{n-1} = \lfloor \hat{x}_{n-1} \rceil$. Continuing thus, we solve for $\hat{x}_{n-2}$ to $\hat{x}_1$, rounding each respectively to obtain $\check{x}^R_{n-2}$ to $\check{x}^R_1$. In general, the i^{th} entries of $\hat{x}$ and $\check{x}^R$ are computed by:

$$\hat{x}_i = \frac{\bar{y}_i - \sum_{j=i+1}^{n} r_{ij}\hat{x}_j}{r_{ii}}, \quad \check{x}^R_i = \lfloor \hat{x}_i \rceil, \quad \text{for } i = n, n-1, ..., 1. \tag{2.4}$$

The IR estimator is

$$\check{x}^R = \lfloor \hat{x} \rceil = \lfloor R^{-1}\bar{y} \rceil = [\lfloor \hat{x}_1 \rceil, \cdots, \lfloor \hat{x}_{n-1} \rceil, \lfloor \hat{x}_n \rceil]^T.$$

Since $v \sim N(0, \sigma^2 I)$ in (1.1), $E\{y\} = Ax$ and $\text{cov}\{y\} = \sigma^2 I$. From $\bar{y} = Q_1^T y$, it follows that $E\{\bar{y}\} = Rx$ and $\text{cov}\{\bar{y}\} = \sigma^2 I$. The RLS estimator $\hat{x} = R^{-1}\bar{y}$ is also random, and we have

$$\begin{aligned} E\{\hat{x}\} &= R^{-1}E\{\bar{y}\} = R^{-1}Rx = x, \\ \text{cov}\{\hat{x}\} &= R^{-1}\text{cov}\{\bar{y}\}R^{-T} = R^{-1}\sigma^2 I R^{-T} = \sigma^2(R^T R)^{-1} \triangleq \Sigma, \end{aligned} \tag{2.5}$$

where $\Sigma \in \mathbb{R}^{n\times n}$ is symmetric positive definite.

Mapping from the RLS Estimator. The pull-in region of an integer estimator is used to find its corresponding success rate. It is defined as a subset of $\mathbb{R}^n$ containing all the real vectors which are mapped to a particular integer vector [23]. We can therefore consider the integer estimation process as a mapping from the RLS solution $\hat{x} \in \mathbb{R}^n$ to an integer vector $\check{x} \in \mathbb{Z}^n$, for ease of understanding. Consequently, in a lot of the GPS literature (see, e.g., [20], [30]) the ILS problem (2.3) is found in its quadratic form, obtained from $\hat{x} = R^{-1}\bar{y}$:

$$||\bar{y} - Rx||_2^2 = ||R(\hat{x} - x)||_2^2 = (\hat{x} - x)^T R^T R(\hat{x} - x).$$

Since $\Sigma = \sigma^2(R^T R)^{-1}$ from (2.5), we have the following minimization problem, equivalent to (2.3):

$$\min_{x \in \mathbb{Z}^n} ||\hat{x} - x||_{\Sigma}^2, \tag{2.6}$$

where $||a||_{\Sigma}^2 = a^T \Sigma^{-1} a$ for $a \in \mathbb{R}^n$.

2.3 Babai Nearest Plane (BNP) Estimation

Another simple integer estimator for x in (2.3), that gives the Babai point, is obtained using the BNP method which, similarly to IR, involves solving an upper triangular system by back substitution. Starting from the n^{th} equation, we obtain $\check{x}_n^B = \left\lfloor \frac{\bar{y}_n}{r_{nn}} \right\rceil$ and then use this integer in the $(n-1)^{th}$ equation to find $\check{x}_{n-1}^B = \left\lfloor \frac{\bar{y}_{n-1} - r_{n-1,n}\check{x}_n^B}{r_{n-1,n-1}} \right\rceil$. We continue thus to solve for $\check{x}_{n-2}^B$ to $\check{x}_1^B$, at each step using the integer estimates found in the previous steps. In general, the i^{th} entry of $\check{x}^B$ is computed by the following:

$$w_i \triangleq \frac{\bar{y}_i - \sum_{j=i+1}^{n} r_{ij}\check{x}_j^B}{r_{ii}}, \quad \check{x}_i^B = \lfloor w_i \rceil, \quad \text{for } i = n, n-1, ..., 1, \tag{2.7}$$

with $w_i \in \mathbb{R}$. From this, the BNP estimator is

$$\check{x}^B = [\check{x}_1^B, \cdots, \check{x}_{n-1}^B, \check{x}_n^B]^T.$$

Given problem (2.6), we can obtain the BNP estimator similarly. The RLS estimator $\hat{x}$ satisfies $\bar{y} = R\hat{x}$. Thus we can rewrite w_i in (2.7) as follows:

$$w_i = \hat{x}_i + \sum_{j=i+1}^{n} \frac{r_{ij}}{r_{ii}}(\hat{x}_j - \check{x}_j^B), \quad \text{for } i = n, n-1, ..., 1, \tag{2.8}$$

or equivalently

$$w = \check{x}^B + D_R^{-1} R(\hat{x} - \check{x}^B), \tag{2.9}$$

where $D_R = \mathrm{diag}(r_{11}, r_{22} ..., r_{nn})$. Equation (2.9) will be used later.

Another form of the BNP estimator commonly found in GPS literature (see, e.g., [20], [22], [30]) is in terms of conditional variances and covariances obtained from the LDL^T decomposition of the general form of the VC-matrix

$$\Sigma = \begin{bmatrix} \sigma^2_{\hat{x}_1} & \cdots & \sigma_{\hat{x}_1\hat{x}_{n-1}} & \sigma_{\hat{x}_1\hat{x}_n} \\ \vdots & \ddots & \vdots & \vdots \\ \sigma_{\hat{x}_{n-1}\hat{x}_1} & \cdots & \sigma^2_{\hat{x}_{n-1}} & \sigma_{\hat{x}_{n-1}\hat{x}_n} \\ \sigma_{\hat{x}_n\hat{x}_1} & \cdots & \sigma_{\hat{x}_n\hat{x}_{n-1}} & \sigma^2_{\hat{x}_n} \end{bmatrix}, \tag{2.10}$$

where $\sigma^2_{\hat{x}_i}$ denotes the variance of $\hat{x}_i$, and $\sigma_{\hat{x}_i\hat{x}_j}$ denotes the covariance between $\hat{x}_i$ and $\hat{x}_j$. For consistency, we consider the L^TDL decomposition instead, to find $\Sigma = L^TDL$ where D is a diagonal matrix and L is unit lower triangular, with entries

$$\begin{aligned} d_{ii} &= \sigma^2_{\hat{x}_{i|\mathcal{I}}} \triangleq \sigma^2_{\hat{x}_i} - \sum_{k=i+1}^{n} \sigma^2_{\hat{x}_{k|\mathcal{K}}\hat{x}_i} \sigma^{-2}_{\hat{x}_{k|\mathcal{K}}}, \qquad (2.11) \\ l_{ij} &= \sigma_{\hat{x}_{i|\mathcal{I}}\hat{x}_j} \sigma^{-2}_{\hat{x}_{i|\mathcal{I}}} \triangleq (\sigma_{\hat{x}_i\hat{x}_j} - \sum_{k=i+1}^{n} \sigma_{\hat{x}_{k|\mathcal{K}}\hat{x}_i} \sigma_{\hat{x}_{k|\mathcal{K}}\hat{x}_j} \sigma^{-2}_{\hat{x}_{k|\mathcal{K}}}) \sigma^{-2}_{\hat{x}_{i|\mathcal{I}}}, \\ &\text{for } i > j, \quad i = n, n-1, ..., 1. \end{aligned}$$

Here, $\mathcal{I}$ denotes the set of indices $\{i+1, ..., n\}$, and $\mathcal{K} = \{k+1, ..., n\}$. We can obtain the BNP estimator as follows. Let $w - x \triangleq L^{-T}(\hat{x} - x)$, to have

$$\begin{aligned} ||\hat{x} - x||^2_{\Sigma} &= (\hat{x} - x)^T L^{-1} D^{-1} L^{-T} (\hat{x} - x) = (w - x)^T D^{-1} (w - x) \\ &= \sum_{i=1}^{n} \sigma^{-2}_{\hat{x}_{i|\mathcal{I}}} (w_i - x_i)^2. \end{aligned} \tag{2.12}$$

We solve the upper triangular system $L^T(w - x) = \hat{x} - x$ for $w \in \mathbb{R}^n$ using back substitution, and find $\check{x}^B \in \mathbb{Z}^n$ that minimizes the sum in (2.12). Clearly, this is achieved by choosing its entries $\check{x}^B_i = \lfloor w_i \rceil$ if w_i has been determined, for i from n to 1. The n^{th} entry of w is equal to $w_n = \hat{x}_n$, and so $\check{x}^B_n = \lfloor w_n \rceil$.

The i^{th} entries of w and $\check{x}^B$ are taken as follows:

$$\begin{aligned} w_i &= \hat{x}_i - \sum_{k=i+1}^{n} \sigma_{\hat{x}_{k|\mathcal{K}}\hat{x}_i} \sigma^{-2}_{\hat{x}_{k|\mathcal{K}}} (w_k - \check{x}^B_k), \\ \check{x}^B_i &= \lfloor w_i \rceil, \quad \text{for } i = n, n-1, ..., 1. \end{aligned} \tag{2.13}$$

It can easily be proven that w_i in (2.13) is equivalent to the w_i in (2.7).

2.4 Integer Least Squares (ILS) Estimation

A typical approach to solving an ILS problem (2.3) involves two phases: a reduction and a search [5]. The purpose of the reduction is to make the search process more efficient. We first introduce the Schnorr and Euchner discrete search strategy [17], in order to motivate the reduction phase, and then briefly discuss the LLL reduction [13].

2.4.1 Schnorr and Euchner Search

Given (2.3), suppose that the optimal x satisfies the following bound [8]:

$$\begin{aligned} &||\bar{y} - Rx||_2^2 < \beta, \\ \text{or} \quad &\sum_{i=1}^{n} (\bar{y}_i - \sum_{j=i}^{n} r_{ij} x_j)^2 < \beta, \end{aligned} \tag{2.14}$$

where β is a constant. This is a hyper-ellipsoid with center $R^{-1}\bar{y}$. We search this ellipsoid to find the optimal solution. Although there are several search strategies, the most often used is the Schnorr and Euchner strategy [1]. First, we define the following non-integer variables:

$$c_i \triangleq \frac{\bar{y}_i - \displaystyle\sum_{j=i+1}^{n} r_{ij} x_j}{r_{ii}}, \quad \text{for } i = n, n-1, ..., 1. \tag{2.15}$$

Then (2.14) can be rewritten as

$$\sum_{i=1}^{n} r_{ii}^2 (x_i - c_i)^2 < \beta. \tag{2.16}$$

This is equivalent to the following n inequalities:

$$\text{level } n\text{:} \quad (x_n - c_n)^2 < \frac{1}{r_{nn}^2}\beta, \tag{2.17}$$

$$\text{level } n-1\text{:} \quad (x_{n-1} - c_{n-1})^2 < \frac{1}{r_{n-1,n-1}^2}[\beta - r_{nn}^2(x_n - c_n)^2], \tag{2.18}$$

$$\vdots$$

$$\text{level } i\text{:} \quad (x_i - c_i)^2 < \frac{1}{r_{ii}^2}[\beta - \sum_{j=i+1}^{n} r_{jj}^2(x_j - c_j)^2], \tag{2.19}$$

$$\vdots$$

$$\text{level } 1\text{:} \quad (x_1 - c_1)^2 < \frac{1}{r_{11}^2}[\beta - \sum_{i=2}^{n} r_{ii}^2(x_i - c_i)^2]. \tag{2.20}$$

Based on these bounds, a search procedure can be developed [8]. First, at level n, we choose $x_n = \lfloor c_n \rceil$. If it does not satisfy bound (2.17), no integer will, thus there is no integer point inside the ellipsoid. This can be avoided by taking the initial β to be large enough. Next, at level $n-1$, we compute c_{n-1} based on the chosen x_n using (2.15) and then choose x_{n-1} to be the nearest integer to c_{n-1}, $x_{n-1} = \lfloor c_{n-1} \rceil$. If this x_{n-1} fails to satisfy bound (2.18), we go back to level n and choose x_n to be the second nearest integer to c_n, and so on. After choosing x_{n-1}, we move to level $n-2$. We continue this procedure until we reach level 1. At this level, we compute c_1 using (2.15) and choose x_1 to be the nearest integer to c_1. If this x_1 fails to satisfy bound (2.20), we must go back to level 2 and choose x_2 to be the next nearest integer to c_2 and so on. Otherwise, with all x_i satisfying bound (2.19) for i from n to 1, we obtain an integer point $\check{x}$ in the ellipsoid [8].

The crucial next step is to shrink the ellipsoid for efficiency by taking a new bound $\beta = ||\bar{y} - R\check{x}||_2^2$. We then search for a new integer point in the ellipsoid by updating $\check{x}$. We go to level 2 to update x_2 by choosing x_2 to be the next nearest integer to c_2. If bound (2.18) is not satisfied with the new integer x_2, we go to level 3 to update the value of x_3 and so on. Otherwise, we go to level 1 to update x_1 and obtain a new integer point; then we update the ellipsoid bound β and repeat the process. Finally, when we end up at level n and fail to find a new integer for x_n that satisfies bound (2.17), the most recently found integer point $\check{x}$ is the optimal solution we seek [8]. This is a depth-first tree search method.

We can set the initial bound β to be ∞, so that the first integer point the search method finds is the Babai point. Alternatively, we can initially set $\beta = ||\bar{y} - R\check{x}^R||_2^2 = ||\bar{y} - R\lfloor \hat{x} \rceil||_2^2$. For more details, see [8].

2.4.2 LLL Reduction

It was found that the order of diagonal entries in R can greatly affect the search speed. In particular, if we have very large $|r_{ii}|$ for small i but very small $|r_{ii}|$ for large i, the so-called search halting problem will be significant. Consider the case where $n = 2$ for simplicity, with $|r_{22}| \ll |r_{11}|$. This implies that the bound (2.17) is loose and therefore quite a number of integers will satisfy this bound. However, the bound (2.18) is very tight, and the likelihood of not being able to find an integer that satisfies this bound is high, thereby increasing the potential of halting. A reduction algorithm usually strives to obtain $|r_{11}| \leq |r_{22}| \leq \cdots \leq |r_{n-1,n-1}| \leq |r_{nn}|$, although this ordering may not always be achievable [1].

The well-known LLL reduction requires that the upper triangular matrix R obtained from A satisfies the following two criteria:

$$|r_{i-1,j}| \leq \frac{1}{2}|r_{i-1,i-1}|, \quad |r_{i-1,i-1}| \leq \delta\sqrt{r_{i-1,i}^2 + r_{ii}^2}, \tag{2.21}$$
$$\text{for} \quad 1 \leq \delta < 2, \quad j = i, ..., n, \quad i = 2, ..., n,$$

where δ is a constant. Then this R is said to be LLL-reduced.

In [36], it was shown that LLL reduction can be cast as a QRZ factorization.

The unimodular transformations in the QRZ factorization (2.1) include integer Gauss transformations, which are used here to reduce the off-diagonal elements of R to meet the first LLL reduction criterion, and permutation matrices, which are used here to reorder the columns of R to meet the second LLL reduction criterion (see [8] for details). Note that the determinant of R, i.e. the product $|r_{11}r_{22}\cdots r_{nn}|$, remains a constant in the reduction process.

2.5 Pull-in Regions

We can partition the continuous space $\mathbb{R}^n$ into subsets, such that each subset is assigned to a particular gridpoint of the discrete space $\mathbb{Z}^n$. We can then consider the estimation process as choosing a particular gridpoint if the RLS estimator lies in its subset. For each integer vector (or gridpoint) $z \in \mathbb{Z}^n$, we assign a non-empty subset $\mathbb{S}_z \subset \mathbb{R}^n$ that contains all the real vectors $\xi \in \mathbb{R}^n$ which get mapped to the integer vector z [23]. This subset $\mathbb{S}_z$ is called the pull-in region of z:

$$\mathbb{S}_z = \{\xi \in \mathbb{R}^n |\ S(\xi) = z\},\ z \in \mathbb{Z}^n, \tag{2.22}$$

where $S : \mathbb{R}^n \to \mathbb{Z}^n$ is a many-to-one map (due to the natures of $\mathbb{R}^n$ and $\mathbb{Z}^n$), meaning that different real vectors may be mapped to the same integer vector. To check if an integer estimator $\check{x}$ obtained equals to the true vector x, therefore, is equivalent to checking if the RLS estimator $\hat{x}$ belongs to the pull-in region of x, i.e. $\check{x} = x$ if and only if $\hat{x} \in \mathbb{S}_x$.

An integer parameter estimator can be expressed as [23]:

$$\check{x} = \sum_{z\in\mathbb{Z}^n} z s_z(\hat{x}),\ \text{where}\ s_z(\hat{x}) = \begin{cases} 1 & \text{if } \hat{x} \in \mathbb{S}_z, \\ 0 & \text{otherwise,} \end{cases} \tag{2.23}$$

and it is said to be admissible if the following three properties hold:

1. Its pull-in regions cover $\mathbb{R}^n$ completely:

$$\bigcup_{z\in\mathbb{Z}^n} \mathbb{S}_z = \mathbb{R}^n. \tag{2.24}$$

2. Its pull-in regions do not overlap:

$$(\text{interior of } \mathbb{S}_z) \cap (\text{interior of } \mathbb{S}_{\tilde{z}}) = \emptyset,\ \forall z, \tilde{z} \in \mathbb{Z}^n, z \neq \tilde{z}. \tag{2.25}$$

3. Its pull-in regions are translationally invariant:

$$\mathbb{S}_z = z + \mathbb{S}_0,\ \forall z \in \mathbb{Z}^n, \tag{2.26}$$

   where $\mathbb{S}_0$ is the pull-in region of the origin of $\mathbb{Z}^n$.

These properties are motivated by the following reasons [23]. Since $\hat{x} \in \mathbb{R}^n$, the subsets should cover $\mathbb{R}^n$ completely to ensure that all real vectors will be mapped to an integer vector. The interiors of these subsets should be disjoint to ensure that the RLS solution is mapped uniquely to one single integer vector, as required. The third property of translational invariance ensures that when the real solution is perturbed by an integer vector, the corresponding integer solution is perturbed by the same amount. In other words, $S(\xi+z) = S(\xi)+z$, $\xi \in \mathbb{R}^n, z \in \mathbb{Z}^n$. This allows application of the integer remove-restore property: $S(\xi - z) + z = S(\xi)$, thus enabling working with the fractional entries of ξ when the complete entries of ξ are too large. This also implies that [23]

$$\begin{aligned}\mathbb{S}_{z+\tilde{z}} &= \{\xi \in \mathbb{R}^n |\ S(\xi) = z + \tilde{z}\} = \{\xi \in \mathbb{R}^n |\ S(\xi) - \tilde{z} = z = S(\xi - \tilde{z})\} \\ &= \{\xi \in \mathbb{R}^n |\ S(\zeta) = z,\ \xi = \zeta + \tilde{z}\} = \mathbb{S}_z + \tilde{z}. \end{aligned} \tag{2.27}$$

Pull-in regions therefore are translated copies of one another. This property can be restated in terms of the pull-in region of the origin $\mathbb{S}_0$ (2.26).

Clearly, how a real vector will be mapped to an integer vector depends on the estimation process used. For instance, using integer rounding in 1D, all real numbers ξ greater than or equal to $z - 0.5$ and less than $z + 0.5$ will get mapped to the integer z, by the definition of rounding. We therefore have different expressions to describe the pull-in regions of the IR, BNP and ILS estimators.

2.5.1 Integer Rounding Estimator

Recall the IR estimator $\check{x}^R = [\lfloor \hat{x}_1 \rceil, \cdots, \lfloor \hat{x}_{n-1} \rceil, \lfloor \hat{x}_n \rceil]^T$ where $\hat{x}_i$ is computed by (2.4). The pull-in region of the integer estimator $\check{x}^R$ is

$$\mathbb{S}^R_{\check{x}^R} = \{\hat{x} \in \mathbb{R}^n | \ |\hat{x}_i - \check{x}^R_i| \leq \frac{1}{2}, \ i = n, n-1, ..., 1\},$$

since each entry $\hat{x}_i$ of the RLS solution is simply rounded to its nearest integer. For each integer vector $z \in \mathbb{Z}^n$, we have the following IR pull-in region [23]:

$$\mathbb{S}^R_z = \{\xi \in \mathbb{R}^n | \ |\xi_i - z_i| \leq \frac{1}{2}, \ i = n, n-1, ..., 1\}. \tag{2.28}$$

2.5.2 Babai Nearest Plane Estimator

Recall the BNP estimator $\check{x}^B = [\lfloor w_1 \rceil, \cdots, \lfloor w_{n-1} \rceil, \lfloor w_n \rceil]^T$ where w_i is computed by (2.7) or (2.8). The pull-in region of the integer estimator $\check{x}^B$ is

$$\mathbb{S}^B_{\check{x}^B} = \{w \in \mathbb{R}^n | \ |w_i - \check{x}^B_i| \leq \frac{1}{2}, \ i = n, n-1, ..., 1\},$$

since each entry w_i is simply rounded to its nearest integer. However, $w - \check{x}^B = D_R^{-1} R(\hat{x} - \check{x}^B)$ so in terms of $\hat{x}$ we have

$$\mathbb{S}^B_{\check{x}} = \{\hat{x} \in \mathbb{R}^n | \ |e_i^T D_R^{-1} R(\hat{x} - \check{x}^B)| \leq \frac{1}{2}, \ i = n, n-1, ..., 1\},$$

where e_i denotes the i^{th} column of I. More generally, if $w - z \triangleq D_R^{-1} R(\xi - z)$, then for each integer vector $z \in \mathbb{Z}^n$,

$$\begin{aligned} \mathbb{S}^B_z &= \{w \in \mathbb{R}^n | \ |w_i - z_i| \leq \frac{1}{2}, \ i = n, n-1, ..., 1\} \\ &= \{\xi \in \mathbb{R}^n | \ |e_i^T D_R^{-1} R(\xi - z)| \leq \frac{1}{2}, \ i = n, n-1, ..., 1\}. \end{aligned} \tag{2.29}$$

Note that (2.29) is identical to (13) given in [23].

2.5.3 Integer Least Squares Estimator

For the optimal solution x to (2.6), $||\hat{x} - x||^2_\Sigma \leq ||\hat{x} - \tilde{x}||^2_\Sigma$, for all $\tilde{x} \in \mathbb{Z}^n$. Therefore, the pull-in region of each $z \in \mathbb{Z}^n$ consists of the collection of all real vectors $\xi \in \mathbb{R}^n$ that are closer to z (in terms of Σ) than to any other integer gridpoint [30]. For each integer vector $z \in \mathbb{Z}^n$,

$$\mathbb{S}^I_z = \{\xi \in \mathbb{R}^n | \; ||\xi - z||^2_\Sigma \leq ||\xi - \tilde{z}||^2_\Sigma, \; \forall \tilde{z} \in \mathbb{Z}^n\}. \tag{2.30}$$

Remarks. The following figures are 2D examples of pull-in regions, taken directly from chapter 3 of [30], and included here to illustrate the different pull-in regions.

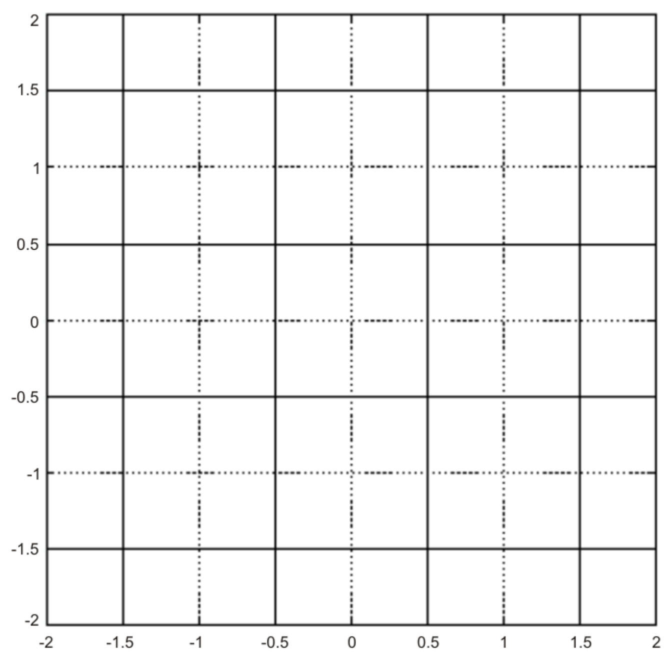

Figure 2–1: 2D IR pull-in regions [30]

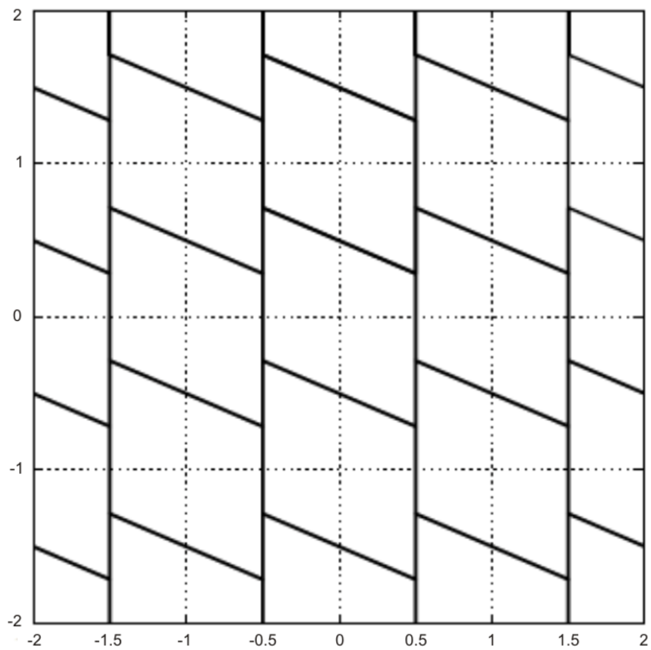

Figure 2–2: 2D BNP pull-in regions [30]

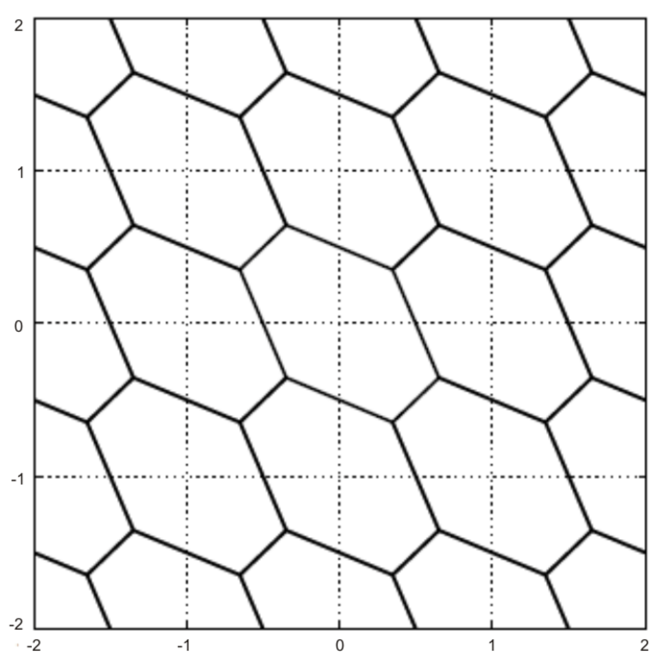

Figure 2–3: 2D ILS pull-in regions [30]

CHAPTER 3
Integer Estimator Validation Using Success Rates

We present the probability distribution of the integer parameter estimator and review the success rates of the IR, BNP and ILS estimators. We also present some success rate properties and results found in the literature. We then discuss partial validation.

3.1 Parameter Probability Distributions

Given a linear model with normally distributed observation data, the linear parameter estimators will also be normally distributed and their uncertainty may be captured by the variance-covariance (VC-) matrix [25]. However, when integer parameters are involved in the estimation process, we have non-normal distributions. To find the uncertainty of the integer parameter estimators, we must determine the parameter probability distributions [25].

Given the linear model (1.1), the RLS estimator $\hat{x}$ of x is normally distributed, i.e. $\hat{x} \sim N(x, \Sigma)$, with mean x and VC-matrix $\Sigma = \sigma^2(R^T R)^{-1}$, see (2.5). The multivariate probability density function (PDF) of $\hat{x}$ is

$$f(\xi) = \frac{1}{\sqrt{\det(\Sigma)(2\pi)^n}} \exp\{-\frac{1}{2}||\xi - x||_{\Sigma}^2\}, \tag{3.1}$$

where $||a||_{\Sigma}^2 = a^T \Sigma^{-1} a$, for $a \in \mathbb{R}^n$.

We can obtain the required distribution of the integer estimator $\check{x}$ from the joint PDF of the real and integer parameters, which we denote as $f_{\hat{x},\check{x}}(\xi, z)$.

Theorem 3.1.1. *The joint distribution of $\hat{x}$ and $\check{x}$ is given as*

$$f_{\hat{x},\check{x}}(\xi, z) = f(\xi) s_z(\xi), \ \xi \in \mathbb{R}^n, z \in \mathbb{Z}^n, \tag{3.2}$$

where $s_z(\xi)$ is the indicator function of the pull-in region $\mathbb{S}_z \subset \mathbb{R}^n$:

$$s_z(\xi) = \begin{cases} 1 & \text{if } \xi \in \mathbb{S}_z, \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

See [25] for the proof of this theorem.

We can recover the marginal distributions of $\hat{x}$ and $\check{x}$ from this joint distribution. The PDF of $\hat{x}$ is

$$\sum_{z \in \mathbb{Z}^n} f_{\hat{x},\check{x}}(\xi, z) = \sum_{z \in \mathbb{Z}^n} f(\xi) s_z(\xi) = f(\xi), \tag{3.4}$$

since $\sum_{z \in \mathbb{Z}^n} s_z(\xi) = 1$ for all $\xi \in \mathbb{R}^n$ by the property that pull-in regions do not overlap, which ensures that a RLS solution is mapped to a unique integer vector. Furthermore, the probability mass function (PMF) of $\check{x}$ is

$$\int_{\mathbb{R}^n} f_{\hat{x},\check{x}}(\xi, z) d\xi = \int_{\mathbb{R}^n} f(\xi) s_z(\xi) d\xi = \int_{\mathbb{S}_z} f(\xi) d\xi. \tag{3.5}$$

Since $f(\xi)$ is the PDF of $\hat{x}$, this integral is equal to the probability $P(\hat{x} \in \mathbb{S}_z)$. Also, by the definition of pull-in regions, $\hat{x} \in \mathbb{S}_z$ is equivalent to $\check{x} = z$, and the following holds [24]:

$$P(\hat{x} \in \mathbb{S}_z) = P(\check{x} = z) = \int_{\mathbb{S}_z} f(\xi) d\xi. \tag{3.6}$$

This is used to find the success rates. This PMF, sometimes referred to in the literature as the integer normal distribution, satisfies the following three useful properties [21] [30].

1. The first is that the distribution is symmetric about x for all admissible integer estimators:

$$P(\check{x} = x - z) = P(\check{x} = x + z), \ \forall z \in \mathbb{Z}^n. \tag{3.7}$$

The proof follows from the fact that the normal distribution is symmetric about x, and can be found in [19] [21].

2. From this, it follows that all admissible estimators are unbiased:

$$E\{\check{x}\} = \sum_{z \in \mathbb{Z}^n} z P(\check{x} = z) = x. \tag{3.8}$$

The proof is given in [19].

3. The third property states that for the ILS estimator, the probability that the integer estimate $\check{x}^I$ coincides with the true but unknown integer vector x is always larger than the probability that it is equal to any other integer vector. In other words, the probability of correct integer estimation is the largest:

$$P(\check{x}^I = x) = \max_{z \in \mathbb{Z}^n} P(\check{x}^I = z). \tag{3.9}$$

The proof for the ILS estimator is given in [21].

3.2 Success Rates

The success rate, which we denote here by P_S, is the probability of correct integer estimation, i.e. the probability that $\check{x}$ coincides with the true x (see, e.g., [11], [23], [24]). From the PMF of $\check{x}$ (3.6), it follows that the success rate can be computed by taking the integral of $f(\xi)$ over $\mathbb{S}_x$, such that [24]:

$$P_S = P(\check{x} = x) = \int_{\mathbb{S}_x} f(\xi) d\xi, \tag{3.10}$$

where $f(\xi)$ is the PDF of $\hat{x}$ (3.1). Note that the success rate depends on the pull-in region of x, and the different estimators have different pull-in regions. The success rate of the IR estimator is

$$P_S^R = P(\check{x}^R = x) = \int_{\mathbb{S}_x^R} f(\xi) d\xi,$$

and the success rates of the BNP and ILS estimators are, correspondingly,

$$P_S^B = P(\check{x}^B = x) = \int\limits_{\mathbb{S}_x^B} f(\xi)d\xi,$$

$$P_S^I = P(\check{x}^I = x) = \int\limits_{\mathbb{S}_x^I} f(\xi)d\xi.$$

Teunissen proves in [23] that the ILS estimator is optimal among the class of admissible integer estimators (which include the IR and BNP estimators).

Theorem 3.2.1. *For any admissible integer estimator $\check{x}$:*

$$P_S^I = P(\check{x}^I = x) \geq P(\check{x} = x) = P_S. \tag{3.11}$$

The proof is based on the definition of the ILS pull-in region. This theorem is important for it justifies using the ILS estimator although it is computationally more expensive than the IR and BNP estimators [23].

3.3 Success Rate of the Integer Rounding Estimator

The success rate P_S^R of the IR estimator is obtained as follows [22]. The pull-in region of x is $\mathbb{S}_x^R = \{\hat{x} \in \mathbb{R}^n | \ |\hat{x}_i - x_i| \leq \frac{1}{2}, \ i = n, ..., 1\}$ by (2.28). For simplicity, we consider the 1D case, i.e. $n = 1$, first. In this case, the VC-matrix of $\hat{x}$ is $\Sigma = \sigma_{\hat{x}}^2$, and

$$\begin{aligned} P_S^R = P(\check{x}^R = x) &= \int\limits_{|\hat{x}-x|\leq\frac{1}{2}} \frac{1}{\sigma_{\hat{x}}\sqrt{2\pi}} \exp\{-\frac{1}{2\sigma_{\hat{x}}^2}(\hat{x}-x)^2\}d\hat{x} \\ &= \int\limits_{-\frac{1}{2}}^{\frac{1}{2}} \frac{1}{\sigma_{\hat{x}}\sqrt{2\pi}} \exp\{-\frac{1}{2\sigma_{\hat{x}}^2}t^2\}dt. \end{aligned}$$

Denoting the cumulative distribution function of the standard normal distribution $N(0,1)$ by $\Phi(\alpha) = \int\limits_{-\infty}^{\alpha} \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}t^2\}dt$, we have

$$P_S^R = \Phi\left(\frac{1}{2\sigma_{\hat{x}}}\right) - \Phi\left(-\frac{1}{2\sigma_{\hat{x}}}\right) = 2\Phi\left(\frac{1}{2\sigma_{\hat{x}}}\right) - 1.$$

Now consider the n dimensional case, with

$$P_S^R = P(\check{x}^R = x) = P(|\hat{x}_i - x_i| \le \frac{1}{2},\ i = n, ..., 1).$$

By the Chain Rule of conditional probabilities, this equals

$$\prod_{i=1}^{n} P\left(|\hat{x}_i - x_i| \le \frac{1}{2} \;\middle|\; |\hat{x}_n - x_n| \le \frac{1}{2},\ ..., |\hat{x}_{i+1} - x_{i+1}| \le \frac{1}{2}\right).$$

Since $\hat{x} \sim N(x, \Sigma)$, the parameter elements are correlated, making exact evaluation of this probability difficult [22]. In the simplest case, where $\Sigma = \mathrm{diag}(\sigma^2_{\hat{x}_1}, \sigma^2_{\hat{x}_2}, ..., \sigma^2_{\hat{x}_n})$ in (2.10), $\hat{x}_j$ for $j = n, ..., 1$ are uncorrelated, and

$$P_S^R = \prod_{i=1}^{n} P(|\hat{x}_i - x_i| \le \frac{1}{2}) = \prod_{i=1}^{n} \left(2\Phi\left(\frac{1}{2\sigma_{\hat{x}_i}}\right) - 1\right).$$

The probability corresponding to the general correlated case is bounded from below by this (see [22]), giving

$$P_S^R = P(\check{x}^R = x) \ge \prod_{i=1}^{n} \left(2\Phi\left(\frac{1}{2\sigma_{\hat{x}_i}}\right) - 1\right). \tag{3.12}$$

3.4 Success Rate of the Babai Nearest Plane Estimator

The success rate P_s^B of the BNP estimator is obtained as follows [22]. From the definition of the pull-in region $\mathbb{S}_x^B$ in (2.29), we have

$$P_S^B = P(\check{x}^B = x) = P(|w_i - x_i| \le \frac{1}{2},\ i = n, ..., 1).$$

By the Chain Rule of conditional probabilities,

$$P_S^B = \prod_{i=1}^{n} P\left(|w_i - x_i| \le \frac{1}{2} \;\middle|\; |w_n - x_n| \le \frac{1}{2},\ ..., |w_{i+1} - x_{i+1}| \le \frac{1}{2}\right).$$

From $w - x = D_R^{-1} R(\hat{x} - x)$ in (2.9) and from (2.5), we find that

$$E\{w - x\} = D_R^{-1} R\ E\{\hat{x} - x\} = D_R^{-1} R \cdot 0 = 0,$$

and also that

$$\begin{aligned}\text{cov}\{w-x\} &= D_R^{-1}R\ \text{cov}\{\hat{x}-x\}(D_R^{-1}R)^T = \sigma^2 D_R^{-1}R\Sigma R^T D_R^{-T} \\ &= \sigma^2 D_R^{-2} = \text{diag}\left(\frac{\sigma^2}{r_{11}^2}, \frac{\sigma^2}{r_{22}^2}, ..., \frac{\sigma^2}{r_{nn}^2}\right). \end{aligned} \tag{3.13}$$

Since $\text{cov}\{w-x\} = \text{cov}\{w\}$ is a diagonal matrix, the entries of w are not correlated. Thus, we have

$$\begin{aligned} P_S^B &= \prod_{i=1}^{n} P(|w_i - x_i| \leq \frac{1}{2}) = \prod_{i=1}^{n} \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{1}{\frac{\sigma}{|r_{ii}|}\sqrt{2\pi}} \exp\{-\frac{1}{2}\frac{t^2}{(\frac{\sigma^2}{r_{ii}^2})}\}dt \\ &= \prod_{i=1}^{n}\left(\Phi\left(\frac{|r_{ii}|}{2\sigma}\right) - \Phi\left(-\frac{|r_{ii}|}{2\sigma}\right)\right) = \prod_{i=1}^{n}\left(2\Phi\left(\frac{|r_{ii}|}{2\sigma}\right) - 1\right). \end{aligned} \tag{3.14}$$

Comparing the Success Rates of the IR and BNP Estimators. Unlike the IR estimator, the BNP estimator computes the i^{th} element $\check{x}_i^B$ using the integer estimates found in the previous steps of the process, i.e. using $\check{x}_{i+1}^B, ..., \check{x}_n^B$. We expect it to have a better success rate than the IR estimator, which does not take any new information into consideration during the estimation process to improve the final integer solution obtained. In fact, Teunissen proved in [22] that

$$P_S^B = P(\check{x}^B = x) \geq P(\check{x}^R = x) = P_S^R. \tag{3.15}$$

3.5 Success Rate of the Integer Least Squares Estimator

Recall the ILS pull-in region of the true integer vector x:

$$\mathbb{S}_x^I = \{\xi \in \mathbb{R}^n |\ ||\xi - x||_\Sigma^2 \leq ||\xi - \tilde{z}||_\Sigma^2,\ \forall \tilde{z} \in \mathbb{Z}^n\}.$$

The success rate integral in (3.10) is difficult to evaluate for the ILS estimator due to this complex integration region [30]. For this reason, we look for lower and upper bounds for P_S^I instead, which would be simpler to evaluate.

3.5.1 BNP Based Lower Bound

From Theorem 3.2.1 we know that the ILS estimator maximizes the success rate [23]. We also know that the success rate of the BNP estimator is at least as high as that of the IR estimator (3.15), and that it can be computed directly and efficiently, as seen in §3.4. Therefore, we can use the BNP success rate as a lower bound on the ILS success rate:

$$P_S^I = P(\check{x}^I = x) \geq \prod_{i=1}^{n} \left(2\Phi\left(\frac{|r_{ii}|}{2\sigma}\right) - 1 \right) = P(\check{x}^B = x) = P_S^B. \qquad (3.16)$$

Remarks. Although the ILS estimator is optimal among the class of admissible integer estimators, there are multiple reasons why we are interested in the BNP estimator. First of all, the Babai point is the first point obtained in the ILS search process, so it is computationally cheaper to find than the ILS solution. Often in practice, the BNP algorithm actually finds the global optimum solution, resulting in a Babai point which coincides with the ILS solution [11]. Also, the BNP success rate can be evaluated exactly, unlike the ILS success rate which requires integration over a complex region, and it provides a good lower bound for the ILS success rate, as simulation results presented in [30] show.

This can be quite useful in practice. As an example, assume the user-defined success rate acceptability threshold is set to 0.85 for a certain application. Before computing the ILS solution, we can first find the Babai point through BNP, evaluate the BNP success rate, and if $P_S^B > 0.85$, we can accept the Babai point, trusting it to be a good enough estimate for the purposes of this application. We can thereby reduce computation costs as there is no need to find the ILS solution in this case.

3.5.2 ADOP Based Upper Bound

We can derive an upper bound on the ILS success rate using a dilution of precision (DOP) measure. DOP measures are simple functions of the appropriate VC-matrices [18]. The GPS ambiguity dilution of precision (ADOP) is defined as a diagnostic that attempts to capture the main characteristics of integer parameter precision [30]. It is defined as

$$ADOP = \left(\sqrt{\det(\Sigma)}\right)^{\frac{1}{n}} = \prod_{i=1}^{n} \sigma_{\hat{x}_{i|I}}^{\frac{1}{n}} . \tag{3.17}$$

This can be obtained by the $L^T DL$ decomposition of the VC-matrix Σ, as in (2.11). One important property of the ADOP is that it can be used to compute the volume of the parameter search space. This volume is a good indicator of the number of gridpoints in the ILS search space [27]. An upper bound for the ILS success rate can be given based on the ADOP.

Theorem 3.5.1. *The ILS success rate is bounded from above by*

$$P_S^I = P(\check{x}^I = x) \leq P\left(\chi^2(n,0) \leq \frac{c_n}{ADOP^2}\right), \tag{3.18}$$

$$\textit{with} \quad c_n = \frac{\left(\frac{n}{2}\Gamma(\frac{n}{2})\right)^{\frac{2}{n}}}{\pi},$$

where $\chi^2(n,0)$ is the central Chi-square distribution with n degrees of freedom, and Γ denotes the gamma function.

This theorem is stated and proved in [24]. The upper bound can be computed using [12]:

$$P(\chi^2(n,0) \leq c) = \begin{cases} 1 - e^{-\frac{c}{2}} \displaystyle\sum_{i=0}^{\frac{1}{2}(n-2)} \frac{(\frac{c}{2})^i}{i!} & \text{if } n \text{ is even,} \\ 2\Phi(\sqrt{c}) - 1 - e^{-\frac{c}{2}} \displaystyle\sum_{i=0}^{\frac{1}{2}(n-3)} \frac{(\frac{c}{2})^{(i+\frac{1}{2})}}{\Gamma(i+\frac{3}{2})} & \text{if } n \text{ is odd,} \end{cases}$$

where $c = \dfrac{c_n}{ADOP^2}$ here.

3.5.3 Integration Region Based Bounds

We can also find lower and upper bounds for the ILS success rate by bounding the integration region $\mathbb{S}_x^I$. For a lower bound, we may replace $\mathbb{S}_x^I$ by a subset $\mathbb{L}_x \subset \mathbb{S}_x^I$ which is completely contained by the pull-in region. For an upper bound, we may replace $\mathbb{S}_x^I$ by an enclosing set $\mathbb{U}_x \supset \mathbb{S}_x^I$ which completely contains the pull-in region. The ILS success rate will then lie in the following interval:

$$P(\hat{x} \in \mathbb{L}_x) \leq P_S^I = P(\hat{x} \in \mathbb{S}_x^I) \leq P(\hat{x} \in \mathbb{U}_x). \tag{3.19}$$

Both regions $\mathbb{L}_x$ and $\mathbb{U}_x$ are chosen by considering the geometry of $\mathbb{S}_x^I$ and also such that they allow easy evaluation of the corresponding probabilities in practice. Details can be found in [21].

Nevertheless, the BNP based lower bound and the ADOP based upper bound are simpler to compute than the integration region based bounds [30]. In addition, they both perform well in most cases [30], providing sharp bounds to the ILS success rate; therefore we focus on extending these particular bounds to box-constrained problems in this thesis.

3.6 Partial Success Rates

We find an estimate $\check{x} \in \mathbb{Z}^n$ to the parameter vector $x \in \mathbb{Z}^n$ using the method of IR, BNP or ILS, and compute the corresponding success rate $P_S = P(\check{x} = x)$ to evaluate the quality of the integer estimate obtained. This is the probability of a simultaneous event $\bigcap_{i=1}^{n}(\check{x}_i = x_i)$, and thereby it tends to decrease as n increases [28]. Therefore it may not always be possible to estimate all n entries of x with a high success rate. Given a success rate acceptability threshold P_T, the goal of partial validation is to find the largest subvector whose success rate is not smaller than P_T. When the overall success rate is less than this threshold value, i.e. $P_S = P(\check{x} = x) < P_T$, we may compute partial success rates P_{PS} of the entries and then fix as integers the largest subset which satisfies $P_{PS} \geq P_T$.

As BNP is a polynomial-time integer estimation method with a success rate P_S^B which can be computed directly, we focus on the partial success rate of the BNP estimator, denoted by P_{PS}^B. Recall the BNP success rate given in §3.4 as follows:

$$P_S^B = P(\check{x}^B = x) = \prod_{i=1}^{n} \left(2\Phi\left(\frac{|r_{ii}|}{2\sigma}\right) - 1 \right).$$

Note that the BNP estimation method finds the entries of $\check{x}^B$ sequentially, from $\check{x}_n^B$ to $\check{x}_1^B$. The probability that the subset of entries of $\check{x}^B$ consisting of $\check{x}_j^B$ to $\check{x}_n^B$, where $i \leq j \leq n$, coincides with the corresponding subset of entries of the true x is obtained by

$$\begin{aligned} P_{PS,j}^B &= P([\check{x}_j^B, \check{x}_{j+1}^B, ..., \check{x}_n^B]^T = [x_j, x_{j+1}, ..., x_n]^T) \\ &= P(|w_k - x_k| \leq \frac{1}{2},\ k = n, ..., j) \\ &= \prod_{k=j}^{n} P(|w_k - x_k| \leq \frac{1}{2} \;\Big|\; |w_n - x_n| \leq \frac{1}{2},\ ..., |w_{k+1} - x_{k+1}| \leq \frac{1}{2}). \end{aligned}$$

See §3.4 for details. Like (3.14), we have

$$P_{PS,j}^B = \prod_{k=j}^{n} P(|w_k - x_k| \leq \frac{1}{2}) = \prod_{k=j}^{n} \left(2\Phi\left(\frac{|r_{kk}|}{2\sigma}\right) - 1 \right). \tag{3.20}$$

How partial validation is used in practice depends on the application itself. For instance, in GPS applications, if the integer ambiguity estimate $\check{x}$ has a low success rate, it is rejected in favour of the RLS solution $\hat{x}$ (2.4) to avoid getting large errors in the position estimates which are dependent on these integer ambiguities. We can use partial validation to find the largest possible subset of $\check{x}$ such that $P_{PS} \geq P_T$, and fix its entries as integers. We can then solve an RLS problem to find real estimates to the remaining entries of the estimate vector, which replace the integer estimates having low success rates. Let $\tilde{x}$ denote the mixed (real and integer) vector outcome of this partial validation process, and let $p_i = 2\Phi\left(\frac{|r_{ii}|}{2\sigma}\right) - 1$, for simplicity of notation. We find $\tilde{x}$ as follows. Initially, we find $P_{PS}^B = p_n$. If $P_{PS}^B < P_T$, we return the original RLS solution $\tilde{x} = \hat{x}$ and stop here. If $P_{PS}^B \geq P_T$, we fix $\tilde{x}_n = \check{x}_n^B$, and continue. We compute $P_{PS}^B = P_{PS}^B \cdot p_{n-1}$, and if $P_{PS}^B < P_T$, we solve an updated RLS problem for $\tilde{x}_{n-1}, ..., \tilde{x}_2, \tilde{x}_1$. If $P_{PS}^B \geq P_T$, we fix $\tilde{x}_{n-1} = \check{x}_{n-1}^B$, and continue. Say that we find $P_{PS}^B = p_n \cdot p_{n-1} \cdots p_j \geq P_T$ but $P_{PS}^B = p_n \cdot p_{n-1} \cdots p_j \cdot p_{j-1} < P_T$. This means that we can successfully fix $\tilde{x}^{(2)} \equiv [\tilde{x}_j, \tilde{x}_{j+1}, ..., \tilde{x}_n]^T = [\check{x}_j^B, \check{x}_{j+1}^B, ..., \check{x}_n^B]^T$. This is the largest possible subset of entries which have a partial success rate higher than the threshold value P_T. The remaining entries of x are computed as follows. Partition x into $\begin{bmatrix} x^{(1)} \\ \tilde{x}^{(2)} \end{bmatrix}$, where $x^{(1)} \in \mathbb{R}^{j-1}$ and $\tilde{x}^{(2)} \in \mathbb{Z}^{n-j+1}$. Similarly, partition $\bar{y}$ into $\begin{bmatrix} \bar{y}^{(1)} \\ \bar{y}^{(2)} \end{bmatrix}$, $\bar{y}^{(1)} \in \mathbb{R}^{j-1}$, $\bar{y}^{(2)} \in \mathbb{R}^{n-j+1}$ and R into $\begin{bmatrix} R_1 & R_{12} \\ 0 & R_2 \end{bmatrix}$ with $R_1 \in \mathbb{R}^{(j-1)\times(j-1)}$, $R_2 \in \mathbb{R}^{(n-j+1)\times(n-j+1)}$, $R_{12} \in \mathbb{R}^{(j-1)\times(n-j+1)}$. We solve the RLS problem:

$$\min_{x^{(1)} \in \mathbb{R}^{j-1}} \left\| \begin{bmatrix} \bar{y}^{(1)} \\ \bar{y}^{(2)} \end{bmatrix} - \begin{bmatrix} R_1 & R_{12} \\ 0 & R_2 \end{bmatrix} \begin{bmatrix} x^{(1)} \\ \tilde{x}^{(2)} \end{bmatrix} \right\|_2^2.$$

This is equivalent to solving

$$\min_{x^{(1)} \in \mathbb{R}^{j-1}} ||(\bar{y}^{(1)} - R_{12}\tilde{x}^{(2)}) - R_1 x^{(1)}||_2^2. \tag{3.21}$$

The RLS estimator is $\tilde{x}^{(1)} = R_1^{-1}(\bar{y}^{(1)} - R_{12}\tilde{x}^{(2)})$.

Theorem 3.6.1. *Let the parameter vector x in* (1.1) *have the partition* $\begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix}$, *where $x^{(1)} \in \mathbb{Z}^{j-1}$ and $x^{(2)} \in \mathbb{Z}^{n-j+1}$ and let the RLS estimator $\hat{x}$ have the* *partition* $\begin{bmatrix} \hat{x}^{(1)} \\ \hat{x}^{(2)} \end{bmatrix}$, *where* $\hat{x}^{(1)} \in \mathbb{R}^{j-1}$ *and* $\hat{x}^{(2)} \in \mathbb{R}^{n-j+1}$. *Suppose* $\tilde{x}^{(2)} \equiv x^{(2)}$. *Then* $\tilde{x}^{(1)}$ *is a more efficient estimator of* $x^{(1)}$ *than* $\hat{x}^{(1)}$ *in the sense that the former has a smaller VC-matrix.*

Proof. Note that

$$E\{\bar{y}^{(1)}\} = R_1 x^{(1)} + R_{12} x^{(2)}.$$

Then

$$\begin{aligned} \tilde{x}^{(1)} &= R_1^{-1}(\bar{y}^{(1)} - R_{12}\tilde{x}^{(2)}) = R_1^{-1}(\bar{y}^{(1)} - R_{12}x^{(2)}) \\ &= R_1^{-1}(\bar{y}^{(1)} - E\{\bar{y}^{(1)}\}) + x^{(1)}. \end{aligned}$$

Thus, $E\{\tilde{x}^{(1)}\} = x^{(1)}$. So $\tilde{x}^{(1)}$ is an unbiased estimator of $x^{(1)}$. Also

$$\begin{aligned} \mathrm{cov}\{\hat{x}\} &= \mathrm{cov}\left\{ \begin{bmatrix} \hat{x}^{(1)} \\ \hat{x}^{(2)} \end{bmatrix} \right\} = \sigma^2 (R^T R)^{-1} \\ &= \sigma^2 \begin{bmatrix} R_1^{-1} & -R_1^{-1}R_{12}R_2^{-1} \\ 0 & R_2^{-1} \end{bmatrix} \begin{bmatrix} R_1^{-T} & 0 \\ -(R_1^{-1}R_{12}R_2^{-1})^T & R_2^{-T} \end{bmatrix}. \end{aligned}$$

Thus

$$\mathrm{cov}\{\hat{x}^{(1)}\} = \sigma^2 (R_1^T R_1)^{-1} + (R_1^{-1} R_{12} R_2)(R_1^{-1} R_{12} R_2)^T. \tag{3.22}$$

But we have

$$\mathrm{cov}\{\tilde{x}^{(1)}\} = R_1^{-1}\mathrm{cov}\{\bar{y}^{(1)}\}R_1^{-T} = R_1^{-1}\sigma^2 I R_1^{-T} = \sigma^2 (R_1^T R_1)^{-1}, \tag{3.23}$$

therefore

$$\mathrm{cov}\{\hat{x}^{(1)}\} - \mathrm{cov}\{\tilde{x}^{(1)}\} = (R_1^{-1} R_{12} R_2)(R_1^{-1} R_{12} R_2)^T,$$

which is symmetric semidefinite. So the conclusion holds. $\square$

CHAPTER 4
Box-constrained Integer Parameter Estimation

We present the box-constrained versions of the IR (i.e. BIR), BNP (i.e. BBNP), and ILS (i.e. BILS) methods of integer estimation and extend the concept of pull-in regions to the corresponding box-constrained estimators.

4.1 Reduced Box-constrained Integer Least Squares Problem

We can transform the original BILS problem (1.6):

$$\min_{x \in \mathbb{B}} ||y - Ax||_2^2, \tag{4.1}$$

where A has full column rank and the unknown integer parameter vector x is constrained to a box $\mathbb{B}$ (1.5), into a new BILS problem by transforming matrix A into an upper triangular matrix R which has good properties that make the search process more efficient [5]. This is accomplished by the QR decomposition of A with column pivoting:

$$AP = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R, \tag{4.2}$$

where $P \in \mathbb{Z}^{n \times n}$ is a permutation matrix, $Q = [Q_1, Q_2] \in \mathbb{R}^{m \times m}$ is orthogonal, and $R \in \mathbb{R}^{n \times n}$ is nonsingular upper triangular [5]. The QR decomposition can be computed using Householder transformations or Givens rotations. The main difference between different reduction strategies in the literature is the permutation matrix P [5].

With the QR factorization (4.2), we have

$$||y - Ax||_2^2 = ||Q_1^T y - RP^T x||_2^2 + ||Q_2^T y||_2^2.$$

Define

$$\bar{y} \triangleq Q_1^T y \in \mathbb{R}^n, \quad \bar{x} \triangleq P^T x, \quad \bar{l} \triangleq P^T l, \quad \bar{u} \triangleq P^T u.$$

We obtain the following reduced BILS problem equivalent to (4.1):

$$\min_{\bar{x} \in \bar{\mathcal{B}}} ||\bar{y} - R\bar{x}||_2^2, \quad \bar{\mathbb{B}} = \{\bar{x} \in \mathbb{Z}^n |\ \bar{l} \leq \bar{x} \leq \bar{u},\ \bar{l} \in \mathbb{Z}^n, \bar{u} \in \mathbb{Z}^n\}. \tag{4.3}$$

For simplicity of notation, we will consider the following problem instead of (4.3) in this thesis:

$$\min_{x \in \mathbb{B}} ||\bar{y} - Rx||_2^2, \quad \mathbb{B} = \{x \in \mathbb{Z}^n |\ l \leq x \leq u,\ l \in \mathbb{Z}^n, u \in \mathbb{Z}^n\}. \tag{4.4}$$

As a final step, therefore, the integer estimators obtained by BIR, BBNP or BILS in the following sections must be left-multiplied by P to obtain the correct estimates to the original x in (4.1). We denote the unconstrained RLS estimator by $\hat{x}$ and a box-constrained integer estimator by $\check{x}^C$. Furthermore, we denote the BIR estimator by $\check{x}^{RC}$, the BBNP estimator by $\check{x}^{BC}$, and the BILS estimator by $\check{x}^{IC}$.

4.2 Box-constrained Integer Rounding (BIR) Estimation

The unconstrained RLS estimator $\hat{x}$ satisfies $\bar{y} = R\hat{x}$, and this upper triangular system can be solved by back substitution, starting from the n^{th} equation. See §2.2 for details. Unlike in unconstrained problems where upon finding $\hat{x}_n$ we choose $\check{x}_n^R = \lfloor \hat{x}_n \rceil$, in box-constrained problems we must take the constraints l_n and u_n into account, since we require $l_n \leq \check{x}_n^C \leq u_n$. If $l_n \leq \lfloor \hat{x}_n \rceil \leq u_n$, we choose $\check{x}_n^{RC} = \lfloor \hat{x}_n \rceil$. Otherwise, we choose $\check{x}_n^{RC}$ to be the nearest integer to $\lfloor \hat{x}_n \rceil$ which satisfies the constraints. Hence, if $\lfloor \hat{x}_n \rceil < l_n$, we choose $\check{x}_n^{RC} = l_n$ and if $\lfloor \hat{x}_n \rceil > u_n$, we choose $\check{x}_n^{RC} = u_n$. Similarly, upon finding $\hat{x}_{n-1}$, we choose $\check{x}_{n-1}^{RC}$ to be the nearest integer to $\lfloor \hat{x}_{n-1} \rceil$ in the constrained interval $[l_{n-1}, u_{n-1}]$. Thus, we choose $\check{x}_{n-1}^{RC} = \lfloor \hat{x}_{n-1} \rceil$ if $l_{n-1} \leq \lfloor \hat{x}_{n-1} \rceil \leq u_{n-1}$, $\check{x}_{n-1}^{RC} = l_{n-1}$ if $\lfloor \hat{x}_{n-1} \rceil < l_{n-1}$, and $\check{x}_{n-1}^{RC} = u_{n-1}$ if $\lfloor \hat{x}_{n-1} \rceil > u_{n-1}$. We continue thus, to find $\check{x}_{n-2}^{RC}, ..., \check{x}_1^{RC}$.

In general, given $\hat{x}_i$ for $i = n, n-1, ..., 1$, as computed by (2.4), the i^{th} entry of $\check{x}^{RC}$ is obtained by:

$$\check{x}_i^{RC} = \begin{cases} \lfloor \hat{x}_i \rceil & \text{if} \quad l_i < \lfloor \hat{x}_i \rceil < u_i, \\ l_i & \text{if} \quad \lfloor \hat{x}_i \rceil \leq l_i, \\ u_i & \text{if} \quad \lfloor \hat{x}_i \rceil \geq u_i. \end{cases} \tag{4.5}$$

From this, the BIR estimator is

$$\check{x}^{RC} = [\check{x}_1^{RC}, \cdots, \check{x}_{n-1}^{RC}, \check{x}_n^{RC}]^T.$$

Mapping from the RLS Estimator. We can again consider the box-constrained integer estimation process as a mapping from the RLS solution $\hat{x} \in \mathbb{R}^n$ to an integer vector $\check{x}^C \in \mathbb{B} \subset \mathbb{Z}^n$. From $\hat{x} = R^{-1}\bar{y}$, we have

$$||\bar{y} - Rx||_2^2 = ||R(\hat{x} - x)||_2^2 = (\hat{x} - x)^T R^T R(\hat{x} - x).$$

Since $\hat{x} \sim N(x, \Sigma)$ with $\Sigma = \sigma^2(R^T R)^{-1}$ by (2.5), we have the following minimization problem, equivalent to (4.4):

$$\min_{x \in \mathbb{B}} ||\hat{x} - x||_{\Sigma}^2, \tag{4.6}$$

where $||a||_{\Sigma}^2 = a^T \Sigma^{-1} a$ for $a \in \mathbb{R}^n$. This form is often used in the GPS literature (see, e.g., [11], [20], [30]).

4.3 Box-constrained Babai Nearest Plane (BBNP) Estimation

We similarly modify the BNP method given in §2.3 to take the box-constraints l and u into account. Solving by back substitution starting from the n^{th} equation, define $w_n \triangleq \dfrac{\bar{y}_n}{r_{nn}}$. We choose $\check{x}_n^{BC}$ to be the nearest integer to $\lfloor w_n \rceil$ in the constrained interval $[l_n, u_n]$, i.e. that satisfies $l_n \leq \check{x}_n^{BC} \leq u_n$. We then use this integer in the $(n-1)^{th}$ equation, where $w_{n-1} \triangleq \dfrac{\bar{y}_{n-1} - r_{n-1,n}\check{x}_n^{BC}}{r_{n-1,n-1}}$.

Again, we choose $\check{x}_{n-1}^{BC}$ to be the nearest integer to $\lfloor w_{n-1} \rceil$ in the constrained interval $[l_{n-1}, u_{n-1}]$. We continue thus to find $\check{x}_{n-2}^{BC}, ..., \check{x}_1^{BC}$, at each step using the integer estimates found in the previous steps. In general, the i^{th} entry of $\check{x}^{BC}$ is computed by the following:

$$w_i \triangleq \frac{\bar{y}_i - \sum_{j=i+1}^{n} r_{ij}\check{x}_j^{BC}}{r_{ii}}, \quad \text{for } i = n, n-1, ..., 1, \tag{4.7}$$

$$\text{and } \check{x}_i^{BC} = \begin{cases} \lfloor w_i \rceil & \text{if} \quad l_i < \lfloor w_i \rceil < u_i, \\ l_i & \text{if} \quad \lfloor w_i \rceil \le l_i, \\ u_i & \text{if} \quad \lfloor w_i \rceil \ge u_i. \end{cases} \tag{4.8}$$

From this, the BBNP estimator is

$$\check{x}^{BC} = [\check{x}_1^{BC}, \cdots, \check{x}_{n-1}^{BC}, \check{x}_n^{BC}]^T.$$

Given problem (4.6), we can obtain the BBNP estimator similarly. The RLS estimator $\hat{x}$ satisfies $\bar{y} = R\hat{x}$. Thus we can rewrite w_i in (4.7) as follows:

$$w_i = \hat{x}_i + \sum_{j=i+1}^{n} \frac{r_{ij}}{r_{ii}}(\hat{x}_j - \check{x}_j^{BC}), \quad \text{for } i = n, n-1, ..., 1, \tag{4.9}$$

or equivalently

$$w = \check{x}^{BC} + D_R^{-1} R(\hat{x} - \check{x}^{BC}), \tag{4.10}$$

where $D_R = \text{diag}(r_{11}, r_{22}, ..., r_{nn})$. We then use (4.8) as well, to obtain the entries of $\check{x}^{BC}$.

4.4 Box-constrained Integer Least Squares (BILS) Estimation

As with unconstrained (ordinary) ILS problems, solving BILS problems involves a reduction phase and a search phase. To make the search process more efficient, a reduction algorithm usually strives for the diagonal elements of matrix R in (4.2) to satisfy $|r_{11}| \le |r_{22}| \le \cdots \le |r_{nn}|$, which may not always be achievable (see [1] for the justification of this ordering). We briefly discuss the ideas of the search and introduce the V-BLAST and SQRD reduction strategies. LLL reduction is usually not used for solving BILS problems as it makes the box-constraint $\mathbb{B}$ very complicated [8].

4.4.1 Search Strategies

The ideas of the Schnorr and Euchner search algorithm for unconstrained problems are described in §2.4. To solve the BILS problem (4.4), the box-constraints must be considered during the search. Details on multiple approaches that modify the unconstrained search in order to take the box-constraints into account can be found in [5]. One such search algorithm, referred to as DEC in [5], is presented in [9], while another search algorithm, referred to as BGBF in [5], is proposed in [4]. Both the DEC and the BGBF strategies are based on the Schnorr and Euchner search. In [5], a new search algorithm, called SEARCH, is provided which uses the advantages of the two algorithms but avoids their drawbacks.

4.4.2 V-BLAST Reduction

The V-BLAST permutation strategy determines the columns of the permuted matrix A from the last column to the first [9]. Let $\mathcal{J}_k$ denote the set of column indices for the columns which have not yet been chosen when the k^{th} column of A is to be determined, for $k = n, n-1, ..., 1$. This strategy chooses the $p(k)^{th}$ column of the original matrix A as the k^{th} column of the permuted matrix A we seek:

$$p(k) = \arg\max_{j \in \mathcal{J}_k} a_j^T [I - A_{k,j}(A_{k,j}^T A_{k,j})^{-1} A_{k,j}^T] a_j, \tag{4.11}$$

where a_j is the j^{th} column of A and $A_{k,j}$ is the $m \times (k-1)$ matrix formed by the columns a_i with $i \in \mathcal{J}_k - \{j\}$.

We can easily show that $a_j^T [I - A_{k,j}(A_{k,j}^T A_{k,j})^{-1} A_{k,j}^T] a_j$ is the square of the Euclidean distance from a_j to the space spanned by the columns of $A_{k,j}$ [5]. Note that in the QR decomposition (4.2), $|r_{kk}|$ is the orthogonal distance from the k^{th} column of AP to the space spanned by the first $k-1$ columns of AP. Thus $a_{p(k)}$ is the column which makes $|r_{kk}|$ maximum over all the remaining columns when we determine the k^{th} column of the permuted A for $k = m, m-1, ..., 1$ [5]. For an efficient algorithm implementing the V-BLAST strategy, refer to [6].

4.4.3 Sorted QR Decomposition

The sorted QR decomposition (SQRD) algorithm, used to find a suboptimal (Babai integer point) solution to the ILS problem, can also be used as a reduction algorithm to speed up the BILS search process [32]. In contrast to the V-BLAST strategy, SQRD determines the columns of the permuted A we seek from the first column to the last, using the modified Gram-Schmidt method [5]. In the k^{th} step of the modified Gram-Schmidt method, the k^{th} column of A is chosen from the remaining $n-k+1$ columns of A such that r_{kk} is smallest, for $k = 1, 2, ..., n$. See [5] for details.

Simulations given in [32] indicate that the Babai point as an estimate of the integer parameter vector obtained by applying the SQRD reduction strategy is slightly less accurate than the Babai point obtained by the V-BLAST strategy. Therefore we expect it to have a lower success rate in general. However, SQRD is computationally more efficient than V-BLAST (see [5] and the references therein).

4.5 Box-constrained Pull-in Regions

The results presented in this section are straightforward extensions of the corresponding results presented in §2.5, found in e.g. [19], [23].

If we think of partitioning $\mathbb{R}^n$ into (different-sized) subsets, such that each subset is assigned to a particular gridpoint of $\mathbb{Z}^n$ that satisfies the box-constraints, we can consider the estimation process as choosing a specific gridpoint if the RLS estimator lies in its subset [19]. If the gridpoint itself lies strictly inside the constraints, the corresponding subset is similar to that of unconstrained problems. Subsets corresponding to gridpoints at the boundaries of the box-constraints are much larger in size, though, since they must include the RLS estimators which would have otherwise (in unconstrained problems) been mapped to gridpoints outside the box-constrained area.

There is an important distinction between the map $S : \mathbb{R}^n \to \mathbb{Z}^n$ of unconstrained problems and the map from the RLS solution to an integer vector in box-constrained problems. For obvious reasons, we require that the box-constrained integer estimators always return integer solutions that satisfy the box-constraints. If we denote this box-constrained map by S^C, then $S^C : \mathbb{R}^n \to \mathbb{B} \subset \mathbb{Z}^n$, where $\mathbb{B}$ is defined in (1.5) as $\mathbb{B} = \{x \in \mathbb{Z}^n |\ l \leq x \leq u,\ l \in \mathbb{Z}^n, u \in \mathbb{Z}^n\}$. From this, it follows that $S^C(\hat{x}) = \check{x}^C$, with $l \leq \check{x}^C \leq u$. The map S^C is a many-to-one map, so different real vectors may be mapped to the same integer vector.

For each integer vector z that satisfies the box-constraints, i.e. $z \in \mathbb{B}$, we can consider a non-empty subset $\mathbb{S}_z^C \subset \mathbb{R}^n$ that contains all the real vectors $\xi \in \mathbb{R}^n$ which get mapped to z. Moreover, no real vector will be mapped to an integer vector outside the box-constraints. This subset $\mathbb{S}_z^C$ is called the box-constrained pull-in region of z:

$$\mathbb{S}_z^C = \begin{cases} \{\xi \in \mathbb{R}^n | S^C(\xi) = z\} & \text{if } z \in \mathbb{B}, \\ \emptyset & \text{if } z \notin \mathbb{B}. \end{cases} \tag{4.12}$$

To check if an integer estimator $\check{x}^C$ obtained equals the true vector x, therefore, is equivalent to checking if the RLS estimator $\hat{x}$ belongs to the pull-in region of x, i.e. $\check{x}^C = x$ if and only if $\hat{x} \in \mathbb{S}_x^C$.

A box-constrained integer parameter estimator can be expressed as:

$$\check{x}^C = \sum_{z \in \mathbb{B}} z s_z^C(\hat{x}), \text{ where } s_z^C(\hat{x}) = \begin{cases} 1 & \text{if } \hat{x} \in \mathbb{S}_z^C \\ \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

For a box-constrained integer estimator to be admissible, its pull-in regions must satisfy the following three properties:

1. Since $\hat{x} \in \mathbb{R}^n$, the subsets should cover $\mathbb{R}^n$ completely, so that all real vectors will be mapped to an integer vector:

$$\bigcup_{z \in \mathbb{B}} \mathbb{S}_z^C = \mathbb{R}^n \tag{4.14}$$

2. The interiors of these subsets should be disjoint so that the real solution is mapped to only one integer vector:

$$(\text{interior of } \mathbb{S}_z^C) \cap (\text{interior of } \mathbb{S}_{\tilde{z}}^C) = \emptyset, \ \forall z, \tilde{z} \in \mathbb{Z}^n, z \neq \tilde{z} \tag{4.15}$$

3. Pull-in regions at the boundaries of the box-constraints are larger than the pull-in regions inside the box-constraints. Therefore, only the pull-in regions $\mathbb{S}_z^C$ that lie strictly inside the box-constraints are translationally invariant. Thus, for $l < z < u$, we have

$$\mathbb{S}_{z+\tilde{z}}^C = \mathbb{S}_z^C + \tilde{z}, \ \forall \tilde{z} \in \mathbb{Z}^n \text{ such that } l < z + \tilde{z} < u \tag{4.16}$$

   Unlike (2.26), this cannot be rewritten in terms of S_0, the pull-in region of the origin of $\mathbb{Z}^n$, because the origin might not satisfy the given constraints.

As in the unconstrained case, we have different expressions to describe the box-constrained pull-in regions of the BIR, BBNP and BILS estimators.

4.5.1 Box-constrained Integer Rounding Estimator

Recall the BIR estimator $\check{x}^{RC} = [\check{x}_1^{RC}, \cdots, \check{x}_{n-1}^{RC}, \check{x}_n^{RC}]^T$, where $\check{x}_i^{RC} = \lfloor \hat{x}_i \rceil$ if $l_i < \lfloor \hat{x}_i \rceil < u_i$, $\check{x}_i^{RC} = l_i$ if $\lfloor \hat{x}_i \rceil \leq l_i$, or $\check{x}_i^{RC} = u_i$ if $\lfloor \hat{x}_i \rceil \geq u_i$, by (4.5). For simplicity, consider the 1D case (i.e. $n = 1$) first. The pull-in region of the integer estimate $\check{x}^{RC}$, when $l < \check{x}^{RC} < u$, is equal to

$$\mathbb{S}_{\check{x}^{RC}}^{RC} = \{\hat{x} \in \mathbb{R} |\ |\hat{x} - \check{x}^{RC}| \leq \frac{1}{2}\} = \{\hat{x} \in \mathbb{R} |\ \hat{x} \in [\check{x}^{RC} - \frac{1}{2}, \check{x}^{RC} + \frac{1}{2})\}.$$

If $\check{x}^{RC} = l$, it is equal to

$$\mathbb{S}_l^{RC} = \{\hat{x} \in \mathbb{R} |\ \hat{x} \in (-\infty, l + \frac{1}{2})\},$$

and similarly, if $\check{x}^{RC} = u$,

$$\mathbb{S}_u^{RC} = \{\hat{x} \in \mathbb{R} |\ \hat{x} \in [u - \frac{1}{2}, \infty)\}.$$

In general, for each integer $z \in \mathbb{Z}$, where z is obtained by rounding $\xi \in \mathbb{R}$:

$$\mathbb{S}_z^{RC} = \begin{cases} \{\xi \in \mathbb{R} |\ |\xi - z| \leq \frac{1}{2}\} & \text{if} \quad l < z < u, \\ \{\xi \in \mathbb{R} |\ \xi \in (-\infty, l + \frac{1}{2})\} & \text{if} \quad z = l, \\ \{\xi \in \mathbb{R} |\ \xi \in [u - \frac{1}{2}, +\infty)\} & \text{if} \quad z = u, \\ \emptyset & \text{if} \quad z < l \text{ or } z > u. \end{cases} \tag{4.17}$$

In other words, all the real numbers which would have been mapped to integers smaller than the lower bound l are now mapped to l itself, while those which would have been mapped to integers larger than the upper bound u are now mapped to u itself. In addition, the pull-in region of any integer outside the constraints is empty, since no real number will be mapped to an integer that does not satisfy the box-constraints.

We can easily verify two of the properties of box-constrained pull-in regions. For each $z \in (l, u)$, we have $\xi \in [z - \frac{1}{2}, z + \frac{1}{2})$, therefore the union of pull-in regions over the area $l < z < u$ is equal to $[l+\frac{1}{2}, u-\frac{1}{2})$. Taking the union of this with the regions at l and u, we have

$$(-\infty, l+\frac{1}{2}) \cup [l+\frac{1}{2}, u-\frac{1}{2}) \cup [u-\frac{1}{2}, +\infty) = (-\infty, +\infty),$$

and so

$$\bigcup_{z\in\mathbb{B}} \mathbb{S}_z^{RC} = \mathbb{R}.$$

Also, it is clear that for any two distinct integers z and $\tilde{z}$, the interiors of $\mathbb{S}_z^C$ and $\mathbb{S}_{\tilde{z}}^C$ are disjoint (empty intersection).

Since the BIR estimator computes each element $\check{x}_i^{RC}$ similarly, we can easily obtain the general form of the BIR pull-in region in n dimensions from (4.17):

$$\mathbb{S}_z^{RC} = \{\xi \in \mathbb{R}^n | \begin{cases} |\xi_i - z_i| \le \frac{1}{2} & \text{if} \quad l_i < z_i < u_i, \\ \xi_i \in (-\infty, l_i + \frac{1}{2}) & \text{if} \quad z_i = l_i, \\ \xi_i \in [u_i - \frac{1}{2}, +\infty) & \text{if} \quad z_i = u_i, \end{cases}$$
$$\text{for} \quad i = n, n-1, ..., 1\}. \tag{4.18}$$

2D Example. We can illustrate this using a simple 2D example. If we let box-constraints $l = [l_1, l_2]^T = [-1, -1]^T$ and $u = [u_1, u_2]^T = [1, 1]^T$, then there is a total of nine different pull-in regions corresponding to integer gridpoints $z = [z_1, z_2]^T$ which satisfy these constraints. Figure 4–1 is based on that of 2D IR pull-in regions in chapter 3 of [30]. Each pull-in region is colored differently for clarity, and is numbered according to the cases below.

1. When $z = [-1, 1]^T$, i.e. $z_1 = l_1,\ z_2 = u_2$, its pull-in region is:

   $S_z = \{\xi \in \mathbb{R}^2 |\ \xi_1 \in (-\infty, l_1 + \frac{1}{2}),\ \xi_2 \in [u_2 - \frac{1}{2}, +\infty)\}$

   $= \{\xi \in \mathbb{R}^2 |\ \xi_1 \in (-\infty, -\frac{1}{2}),\ \xi_2 \in [\frac{1}{2}, +\infty)\}$.

   This means that any RLS solution ξ with $\xi_1 < -\frac{1}{2}$ and $\xi_2 \ge \frac{1}{2}$ will be

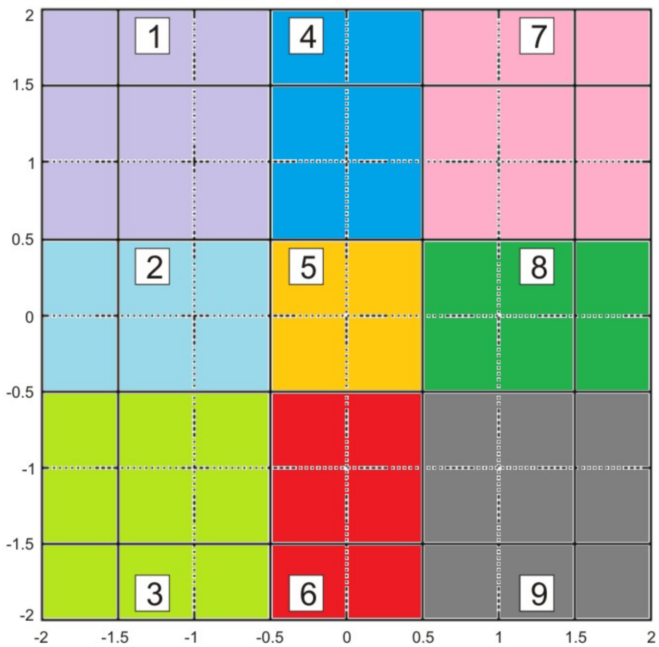

Figure 4–1: 2D BIR pull-in regions (based on [30])

mapped to the integer point $[-1, 1]^T$ using BIR.

Similarly, the following statements hold.

2. When $z = [-1, 0]^T$, $S_z = \{\xi \in \mathbb{R}^2 |\ \xi_1 \in (-\infty, -\frac{1}{2}),\ \xi_2 \in [-\frac{1}{2}, \frac{1}{2})\}$.
3. When $z = [-1, -1]^T$, $S_z = \{\xi \in \mathbb{R}^2 |\ \xi_1 \in (-\infty, -\frac{1}{2}),\ \xi_2 \in (-\infty, -\frac{1}{2})\}$.
4. When $z = [0, 1]^T$, $S_z = \{\xi \in \mathbb{R}^2 |\ \xi_1 \in [-\frac{1}{2}, \frac{1}{2}),\ \xi_2 \in [\frac{1}{2}, +\infty)\}$.
5. When $z = [0, 0]^T$, $S_z = \{\xi \in \mathbb{R}^2 |\ \xi_1 \in [-\frac{1}{2}, \frac{1}{2}),\ \xi_2 \in [-\frac{1}{2}, \frac{1}{2})\}$.
6. When $z = [0, -1]^T$, $S_z = \{\xi \in \mathbb{R}^2 |\ \xi_1 \in [-\frac{1}{2}, \frac{1}{2}),\ \xi_2 \in (-\infty, \frac{1}{2})\}$.
7. When $z = [1, 1]^T$, $S_z = \{\xi \in \mathbb{R}^2 |\ \xi_1 \in [\frac{1}{2}, +\infty),\ \xi_2 \in [\frac{1}{2}, +\infty)\}$.
8. When $z = [1, 0]^T$, $S_z = \{\xi \in \mathbb{R}^2 |\ \xi_1 \in [\frac{1}{2}, +\infty),\ \xi_2 \in [-\frac{1}{2}, \frac{1}{2})\}$.
9. When $z = [1, -1]^T$, $S_z = \{\xi \in \mathbb{R}^2 |\ \xi_1 \in [\frac{1}{2}, +\infty),\ \xi_2 \in (-\infty, \frac{1}{2})\}$.

We can similarly derive pull-in regions for the BBNP and BILS estimators.

4.5.2 Box-constrained Babai Nearest Plane Estimator

Recall the BBNP estimator $\check{x}^{BC} = [\check{x}_1^{BC}, \cdots, \check{x}_{n-1}^{BC}, \check{x}_n^{BC}]^T$, where $\check{x}_i^{BC} = \lfloor w_i \rceil$ if $l_i < \lfloor w_i \rceil < u_i$, $\check{x}_i^{BC} = l_i$ if $\lfloor w_i \rceil \le l_i$, or $\check{x}_i^{BC} = u_i$ if $\lfloor w_i \rceil \ge u_i$ by (4.8), and w_i is computed by (4.7) or (4.9). The pull-in region of the integer estimate $\check{x}^{BC}$ is

$$\mathbb{S}^{BC}_{\check{x}^{BC}} = \{w \in \mathbb{R}^n | \begin{cases} |w_i - \check{x}_i^{BC}| \leq \frac{1}{2} & \text{if} \quad l_i < \check{x}_i^{BC} < u_i, \\ w_i \in (-\infty, l_i + \frac{1}{2}) & \text{if} \quad \check{x}_i^{BC} = l_i, \\ w_i \in [u_i - \frac{1}{2}, +\infty) & \text{if} \quad \check{x}_i^{BC} = u_i, \end{cases}$$
$$\text{for} \quad i = n, n-1, ..., 1\}. \tag{4.19}$$

Since $w = \check{x}^{BC} + D_R^{-1} R(\hat{x} - \check{x}^{BC})$ from (4.10), we have

$$\mathbb{S}^{BC}_{\check{x}^{BC}} = \{\hat{x} \in \mathbb{R}^n | \begin{cases} |e_i^T D_R^{-1} R(\hat{x} - \check{x}^{BC})| \leq \frac{1}{2} & \text{if} \quad l_i < \check{x}_i^{BC} < u_i, \\ \hat{x}_i \in (-\infty, l_i + \frac{1}{2} - \sum_{j=i+1}^{n} \frac{r_{ij}}{r_{ii}} (\hat{x}_j - \check{x}_j^{BC})\ ) & \text{if} \quad \check{x}_i^{BC} = l_i, \\ \hat{x}_i \in [u_i - \frac{1}{2} - \sum_{j=i+1}^{n} \frac{r_{ij}}{r_{ii}} (\hat{x}_j - \check{x}_j^{BC}), +\infty) & \text{if} \quad \check{x}_i^{BC} = u_i, \end{cases}$$
$$\text{for} \quad i = n, n-1, ..., 1\}.$$

We want a general expression for each integer vector $z \in \mathbb{Z}^n$, so if $w - z \triangleq D^{-1}R(\xi - z)$, we have an expression like (4.19) for $\mathbb{S}_z^{BC}$, from which we obtain

$$\mathbb{S}^{BC}_{z} = \{\xi \in \mathbb{R}^n | \begin{cases} |e_i^T D_R^{-1} R(\xi - z)| \leq \frac{1}{2} & \text{if} \quad l_i < z_i < u_i, \\ \xi_i \in (-\infty, l_i + \frac{1}{2} - \sum_{j=i+1}^{n} \frac{r_{ij}}{r_{ii}} (\xi_j - z_j)\ ) & \text{if} \quad z_i = l_i, \\ \xi_i \in [u_i - \frac{1}{2} - \sum_{j=i+1}^{n} \frac{r_{ij}}{r_{ii}} (\xi_j - z_j), +\infty) & \text{if} \quad z_i = u_i, \end{cases}$$
$$\text{for} \quad i = n, n-1, ..., 1\}. \tag{4.20}$$

Figure 4–2 is a 2D example of BBNP pull-in regions with box-constraints $l = [-1, -1]^T$ and $u = [1, 1]^T$. It is based on that of 2D BNP pull-in regions in chapter 3 of [30]. Each pull-in region is colored differently for clarity.

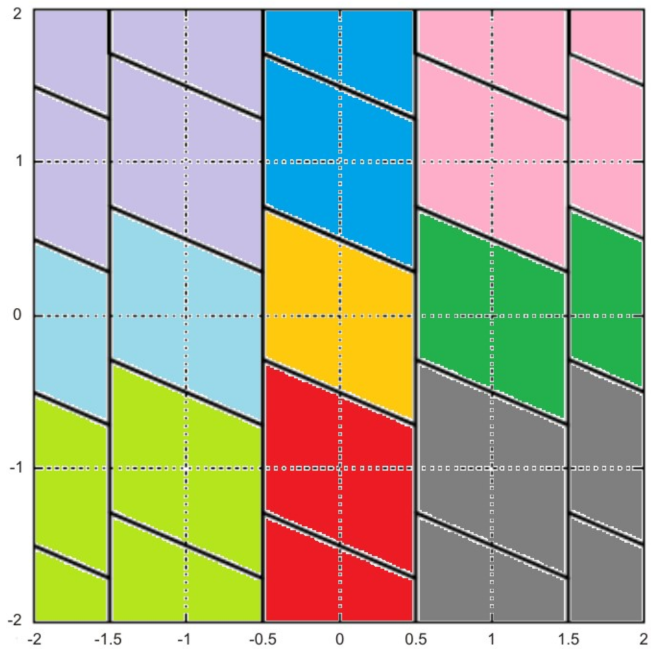

Figure 4–2: 2D BBNP pull-in regions (based on [30])

4.5.3 Box-constrained Integer Least Squares Estimator

The general BILS pull-in region for $z \in \mathbb{B}$ is

$$\mathbb{S}_z^{IC} = \{\xi \in \mathbb{R}^n | \ \ ||\xi - z||_\Sigma^2 \leq ||\xi - \tilde{z}||_\Sigma^2, \ \forall \tilde{z} \in \mathbb{B}\} \tag{4.21}$$

Figure 4–3 is a 2D example of BILS pull-in regions with box-constraints $l = [-1, -1]^T$ and $u = [1, 1]^T$. It is based on that of 2D ILS pull-in regions in chapter 3 of [30]. Each pull-in region is colored differently for clarity.

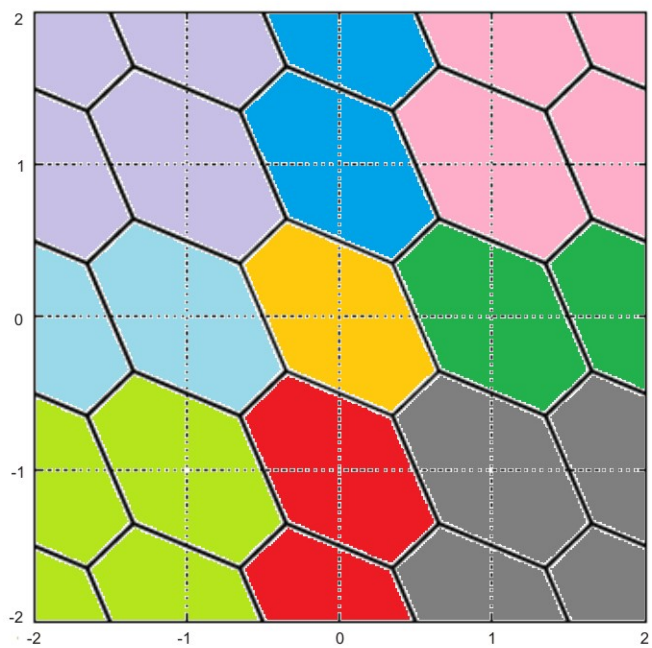

Figure 4–3: 2D BILS pull-in regions (based on [30])

CHAPTER 5
Box-constrained Integer Estimator Validation Using Success Rates

We extend the concept of success rates presented in chapter 3 to the BIR, BBNP and BILS estimators, along with some success rate results. We then apply the extended results, in particular for partial validation, to improve the efficiency of the BILS estimation process. Numerical simulations results are presented to support our findings.

5.1 Parameter Probability Distributions

Given the linear model (1.1), the RLS estimator $\hat{x}$ of x is normally distributed, i.e. $\hat{x} \sim N(x, \Sigma)$, with mean x and VC-matrix $\Sigma = \sigma^2 (R^T R)^{-1}$, see (2.5). The multivariate probability density function (PDF) of $\hat{x}$ is:

$$f(\xi) = \frac{1}{\sqrt{\det(\Sigma)(2\pi)^n}} \exp\{-\frac{1}{2}||\xi - x||_{\Sigma}^2\}. \tag{5.1}$$

We can obtain the required distribution of the box-constrained integer estimator $\check{x}^C$ from the joint PDF of the real and integer parameters, which we denote here as $f_{\hat{x},\check{x}^C}(\xi, z)$.

Theorem 5.1.1. *The joint distribution of $\hat{x}$ and $\check{x}^C$ is given as*

$$f_{\hat{x},\check{x}^C}(\xi, z) = f(\xi) s_z^C(\xi), \ \xi \in \mathbb{R}^n, z \in \mathbb{Z}^n, \tag{5.2}$$

where $s_z^C(\xi)$ is the indicator function of the pull-in region $S_z^C \subset \mathbb{R}^n$:

$$s_z^C(\xi) = \begin{cases} 1 & \textit{if } \xi \in \mathbb{S}_z^C, \\ 0 & \textit{otherwise.} \end{cases} \tag{5.3}$$

This theorem is an extension of Theorem 3.1.1, and its proof is a simple extension of that theorem's proof, given in [25].

Proof. Let $f_{\hat{x}|\check{x}^C}(\xi|z)$ be the conditional distribution of $\hat{x}$ given $\check{x}^C = z$. Then, for arbitrary $\Omega \subset \mathbb{R}^n$

$$P(\hat{x} \in \Omega | \check{x}^C = z) = \int_{\Omega} f_{\hat{x}|\check{x}^C}(\xi|z) d\xi. \tag{5.4}$$

On the other hand,

$$P(\hat{x} \in \Omega | \check{x}^C = z) = P(\hat{x} \in \Omega | \hat{x} \in \mathbb{S}_z^C) \ = \frac{P(\hat{x} \in \Omega, \hat{x} \in \mathbb{S}_z^C)}{P(\hat{x} \in \mathbb{S}_z^C)},$$

since $\check{x}^C = z$ if and only if $\hat{x} \in \mathbb{S}_z^C$. Moreover, $P(\hat{x} \in \Omega, \hat{x} \in \mathbb{S}_z^C) = \int\limits_{\Omega \cap \mathbb{S}_z^C} f(\xi) d\xi$, so

$$P(\hat{x} \in \Omega | \check{x}^C = z) = \frac{\int\limits_{\Omega \cap \mathbb{S}_z^C} f(\xi) d\xi}{P(\hat{x} \in \mathbb{S}_z^C)} = \int_{\Omega} \frac{f(\xi) s_z^C(\xi) d\xi}{P(\hat{x} \in \mathbb{S}_z^C)}. \tag{5.5}$$

It therefore follows from (5.4) and (5.5) that

$$f_{\hat{x}|\check{x}^C}(\xi|z) P(\hat{x} \in \mathbb{S}_z^C) = f(\xi) s_z^C(\xi) = f_{\hat{x},\check{x}^C}(\xi, z). \tag{5.6}$$

□

We can recover the marginal distributions of $\hat{x}$ and $\check{x}^C$ from this joint distribution. The PDF of $\hat{x}$ is

$$\sum_{z \in \mathbb{Z}^n} f_{\hat{x},\check{x}^C}(\xi, z) = \sum_{z \in \mathbb{Z}^n} f(\xi) s_z^C(\xi) = f(\xi), \tag{5.7}$$

since $\sum\limits_{z \in \mathbb{Z}^n} s_z^C(\xi) = 1$ for all $\xi \in \mathbb{R}^n$ by the property that box-constrained pull-in regions do not overlap, which ensures that a RLS solution is mapped to a unique integer vector (satisfying the box-constraints).

Furthermore, the probability mass function (PMF) of $\check{x}^C$ is

$$\int_{\mathbb{R}^n} f_{\hat{x},\check{x}^C}(\xi, z) d\xi = \int_{\mathbb{R}^n} f(\xi) s_z^C(\xi) d\xi = \int_{\mathbb{S}_z^C} f(\xi) d\xi = P(\hat{x} \in \mathbb{S}_z^C), \tag{5.8}$$

and this is equal to the probability $P(\check{x}^C = z)$ by the equivalence of $\hat{x} \in \mathbb{S}_z^C$ and $\check{x}^C = z$. In §3.1, we presented three properties which hold for the integer normal distribution (i.e. the PMF of $\check{x}$). We investigate whether these properties also hold for the box-constrained distribution.

1. In the unconstrained case, the PMF of $\check{x}$ is symmetric about x for all admissible integer estimators, i.e. $P(\check{x} = x - z) = P(\check{x} = x + z)$, $\forall z \in \mathbb{Z}^n$. Clearly this property will not always hold for the PMF of $\check{x}^C$, due to the box-constraints.

   Example. Given true $x \in \mathbb{Z}$, box-constraints l and $u = l + 4$, with $x = u$, and $z = 2$, we have

$$
\begin{aligned}
0 < & P(\check{x}^C = x - z = l + 2 = u - 2) < 1, \\
& P(\check{x}^C = x + z = u + 2) = 0,
\end{aligned}
$$

   since $l \leq \check{x}^C \leq u$.

2. In the unconstrained case, all admissible estimators are unbiased, i.e. $E\{\check{x}\} = \sum_{z \in \mathbb{Z}^n} z P(\check{x} = z) = x$. However, the box-constrained integer estimators are usually biased, given the fact that the PMF of $\check{x}^C$ is not always symmetric about x, and that inequality-restricted real estimators are generally biased [29].

3. In the unconstrained case, for the ILS estimator, the probability of correct estimation is the largest, i.e. $P(\check{x}^I = x) = \max_{z \in \mathbb{Z}^n} P(\check{x}^I = z)$. In the box-constrained case, we cannot get a similar result for the BILS estimator. We give a counter example to illustrate this. Note that for this counter example, we use the success rate of the BBNP estimator, which in one dimension is the same as the success rate of the BILS estimator. Details on the BBNP success rate are given in §5.4.

   Example. Given the linear model

$$
y = \frac{1}{2}x + v,
$$

with $v \in \mathbb{R}$ following the normal distribution, i.e. $v \sim N(0,1)$. Since the VC-matrix $\Sigma = \sigma^2(A^TA)^{-1} = 4$, $R = \frac{1}{2}$. Given true $x = 2$, box-constraints $l = 1$ and $u = 3$, and $z = 1$, we have

$$P(\check{x}^{IC} = x) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{1}{2\sqrt{(2\pi)}} \exp\{-\frac{1}{8}t^2\}dt = 2\Phi\left(\frac{1}{4}\right) - 1 = 0.1974126514,$$

$$P(\check{x}^{IC} = z) = \int_{-\infty}^{\frac{3}{2}} \frac{1}{2\sqrt{(2\pi)}} \exp\{-\frac{1}{2}\frac{(t-2)^2}{4}\}dt = 0.4012936743,$$

by equation (5.11). Thus, in this example, $P(\check{x}^{IC} = x) < P(\check{x}^{IC} = z)$.

5.2 Box-constrained Success Rates

The box-constrained success rate, which we denote here by P_S^C, is the probability that the box-constrained integer estimate $\check{x}^C$ obtained coincides with the true integer vector x. From the PMF of $\check{x}^C$ (5.8), it follows that the success rate can be computed by taking the integral of $f(\xi)$, the PDF of the RLS estimator $\hat{x}$, over $\mathbb{S}_x^C$, the box-constrained pull-in region of x:

$$P_S^C = P(\check{x}^C = x) = \int_{\mathbb{S}_x^C} f(\xi)d\xi. \tag{5.9}$$

Since the success rate depends on the pull-in region of x, and the different estimators have different pull-in regions, the success rates of the BIR, BBNP and BILS estimators are, correspondingly,

$$P_S^{RC} = P(\check{x}^{RC} = x) = \int_{\mathbb{S}_x^{RC}} f(\xi)d\xi,$$
$$P_S^{BC} = P(\check{x}^{BC} = x) = \int_{\mathbb{S}_x^{BC}} f(\xi)d\xi,$$
$$P_S^{IC} = P(\check{x}^{IC} = x) = \int_{\mathbb{S}_x^{IC}} f(\xi)d\xi.$$

In the unconstrained case, Theorem 3.2.1 states that the ILS estimator is optimal among all admissible integer estimators, including the IR and BNP estimators. However, we cannot extend this result to box-constrained problems. The BILS estimator cannot be shown to be optimal among all box-constrained admissible estimators. We give a counter example to illustrate this.

Example. Let $R = \begin{bmatrix} 0.3 & -0.2 \\ 0 & 0.3 \end{bmatrix}$, true vector $x = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$, box-constraints $l = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $u = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$, and $\sigma = 1$.

By (4.20), the pull-in region of the BBNP estimator is

$$\mathbb{S}^{BC} = \{\xi \in \mathbb{R}^2 | \xi_1 \geq \frac{2}{3}\xi_2 + \frac{1}{6}, \xi_2 \geq \frac{3}{2}\}.$$

Note that ξ here is the RLS estimator. By (4.21), the pull-in region of the BILS estimator is

$$\begin{aligned} \mathbb{S}^{IC} = \{\xi \in \mathbb{R}^2 | &[0.3(\xi_1 - 2) - 0.2(\xi_2 - 2)]^2 + 0.09(\xi_2 - 2)^2 \leq \\ &[0.3(\xi_1 - \alpha) - 0.2(\xi_2 - \beta)]^2 + 0.09(\xi_2 - \beta)^2, \forall \alpha = 1, 2, \beta = 1, 2\} \\ = \{\xi \in \mathbb{R}^2 | &[3(\xi_1 - 2) - 2(\xi_2 - 2)]^2 + 9(\xi_2 - 2)^2 \leq \\ &[3(\xi_1 - \alpha) - 2(\xi_2 - \beta)]^2 + 9(\xi_2 - \beta)^2, \forall \alpha = 1, 2, \beta = 1, 2\}. \end{aligned}$$

Equivalently, ξ satisfies the following inequalities:

$$\begin{aligned} 6(3\alpha - 2\beta - 2)\xi_1 + 2(-6\alpha + 13\beta - 14)\xi_2 \leq (3\alpha - 2\beta)^2 + 9\beta^2 - 40&, \\ \text{and for } \alpha = 1, \beta = 1, \quad \xi_1 \geq 5 - \frac{7}{3}\xi_2&, \\ \text{for } \alpha = 1, \beta = 2, \quad \xi_1 \geq \frac{1}{6} + \frac{2}{3}\xi_2&, \\ \text{for } \alpha = 2, \beta = 1, \quad \xi_1 \leq -\frac{5}{4} + \frac{13}{6}\xi_2&, \\ \text{for } \alpha = 2, \beta = 2, \quad \xi \in \mathbb{R}^2&. \end{aligned}$$

It is easy to verify that

$$S^{IC} = \{\xi \in \mathbb{R}^2 | 5 - \frac{7}{3}\xi_2 \leq \xi_1 \leq -\frac{5}{4} + \frac{13}{6}\xi_2,\ \frac{25}{18} \leq \xi_2 \leq \frac{29}{18}\}$$
$$\cup \{\xi \in \mathbb{R}^2 | \frac{1}{6} + \frac{2}{3}\xi_2 \leq \xi_1 \leq -\frac{5}{4} + \frac{13}{6}\xi_2,\ \xi_2 \geq \frac{29}{18}\}.$$

Since $\xi \sim N(x, \Sigma)$ with $\Sigma = \sigma^2(R^T R)^{-1}$ by (2.5), we have

$$P_S^{BC} = \int_{S^{BC}} \frac{1}{\sqrt{\det(\Sigma)(2\pi)^2}} \exp\{-\frac{1}{2}[(0.3\xi_1 - 0.2\xi_2 - 0.2)^2 + 0.09(\xi_2 - 2)^2)]\} d\xi$$
$$= 0.3131719615,$$
$$P_S^{IC} = \int_{S^{IC}} \frac{1}{\sqrt{\det(\Sigma)(2\pi)^2}} \exp\{-\frac{1}{2}[(0.3\xi_1 - 0.2\xi_2 - 0.2)^2 + 0.09(\xi_2 - 2)^2)]\} d\xi$$
$$= 0.2268633995,$$

using equations (5.1) and (5.9). Thus, in this example, $P_S^{IC} < P_S^{BC}$.

5.3 Success Rate of the Box-constrained Integer Rounding Estimator

The success rate P_S^{RC} of the BIR estimator is obtained as follows. Define $\mathbb{B}_i = \{\hat{x}_i \in \mathbb{R} |\ |\hat{x}_i - x_i| \leq \frac{1}{2}$ if $l_i < x_i < u_i,\ \hat{x}_i \in (-\infty, l_i + \frac{1}{2})$ if $x_i = l_i,\ \hat{x}_i \in [u_i - \frac{1}{2}, +\infty)$ if $x_i = u_i\}$. From (4.18), the box-constrained pull-in region of x is $\mathbb{S}_x^{RC} = \{\hat{x} \in \mathbb{R}^n |\ \hat{x}_i \in \mathbb{B}_i,$ for $i = n, n-1, ..., 1\}$. For simplicity, we consider the 1D case, i.e. $n = 1$, first. In this case, the VC-matrix of $\hat{x}$ is $\Sigma = \sigma_{\hat{x}}^2$. There are three possible success rates, depending on the value of x with respect to the constraints l and u. When $l < x < u$, we have

$$P(\check{x}^{RC} = x | l < x < u) = \int_{|\hat{x} - x| \leq \frac{1}{2}} \frac{1}{\sigma_{\hat{x}}\sqrt{2\pi}} \exp\{-\frac{1}{2\sigma_{\hat{x}}^2}(\hat{x} - x)^2\} d\hat{x}$$
$$= \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{1}{\sigma_{\hat{x}}\sqrt{2\pi}} \exp\{-\frac{1}{2\sigma_{\hat{x}}^2}t^2\} dt = 2\Phi\left(\frac{1}{2\sigma_{\hat{x}}}\right) - 1,$$

where $\Phi(\alpha) = \int\limits_{-\infty}^{\alpha} \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}t^2\}dt$ denotes the cumulative distribution function of the standard normal distribution $N(0,1)$. Similarly, if $x = l$, we have

$$\begin{aligned} P(\check{x}^{RC} = x | x = l) &= \int\limits_{-\infty}^{l+\frac{1}{2}} \frac{1}{\sigma_{\hat{x}}\sqrt{2\pi}} \exp\{-\frac{1}{2\sigma_{\hat{x}}^2}(\hat{x}-x)^2\}d\hat{x} \\ &= \int\limits_{-\infty}^{\frac{1}{2}} \frac{1}{\sigma_{\hat{x}}\sqrt{2\pi}} \exp\{-\frac{1}{2\sigma_{\hat{x}}^2}t^2\}dt = \Phi\left(\frac{1}{2\sigma_{\hat{x}}}\right), \end{aligned}$$

and if $x = u$, we have

$$\begin{aligned} P(\check{x}^{RC} = x | x = u) &= \int\limits_{u-\frac{1}{2}}^{\infty} \frac{1}{\sigma_{\hat{x}}\sqrt{2\pi}} \exp\{-\frac{1}{2\sigma_{\hat{x}}^2}(\hat{x}-x)^2\}d\hat{x} \\ &= \int\limits_{-\frac{1}{2}}^{\infty} \frac{1}{\sigma_{\hat{x}}\sqrt{2\pi}} \exp\{-\frac{1}{2\sigma_{\hat{x}}^2}t^2\}dt = \Phi\left(\frac{1}{2\sigma_{\hat{x}}}\right). \end{aligned}$$

Now consider the n dimensional case, with

$$P_S^{RC} = P(\check{x}^{RC} = x) = P(\hat{x}_i \in \mathbb{B}_i, \ i = n, n-1, ..., 1).$$

By the Chain Rule of conditional probabilities, this is equal to:

$$P_S^{RC} = \prod_{i=1}^{n} P(\hat{x}_i \in \mathbb{B}_i | \hat{x}_n \in \mathbb{B}_n, ..., \hat{x}_{i+1} \in \mathbb{B}_{i+1}).$$

Since $\hat{x} \sim N(x, \Sigma)$, the parameter elements are correlated, making exact evaluation of this probability difficult [22]. In the simplest case, where $\Sigma = \text{diag}(\sigma_{\hat{x}_1}^2, \sigma_{\hat{x}_2}^2 ..., \sigma_{\hat{x}_n}^2)$ in (2.10), $\hat{x}_j$ for $j = n, ..., 1$ are uncorrelated, and

$$\begin{aligned} P_S^{RC} &= P(\check{x}^{RC} = x) = \prod_{i=1}^{n} P(\hat{x}_i \in \mathbb{B}_i) \\ &= \prod_{i=1}^{n} \begin{cases} \left(2\Phi\left(\frac{1}{2\sigma_{\hat{x}_i}}\right) - 1\right) & \text{if } l_i < x_i < u_i, \\ \Phi\left(\frac{1}{2\sigma_{\hat{x}_i}}\right) & \text{if } x_i = l_i \text{ or } x_i = u_i. \end{cases} \end{aligned}$$

In the general correlated case, we have

$$P_S^{RC} = P(\check{x}^{RC} = x) = \int\limits_{\mathbb{S}_x^{RC}} f(\xi) d\xi \tag{5.10}$$

$$= \int\limits_{\xi_1 \in \mathbb{B}_1} \cdots \int\limits_{\xi_n \in \mathbb{B}_n} \frac{1}{\sqrt{\det(\Sigma)(2\pi)^n}} \exp\{-\frac{1}{2}||\xi - x||_\Sigma^2\} d\xi_n \cdots d\xi_1.$$

5.4 Success Rate of the Box-constrained Babai Nearest Plane Estimator

The success rate P_S^{BC} of the BBNP estimator is obtained as follows. Define $\mathbb{B}_i = \{w_i \in \mathbb{R} |\ |w_i - x_i| \le \frac{1}{2}$ if $l_i < x_i < u_i,\ w_i \in (-\infty, l_i + \frac{1}{2})$ if $x_i = l_i,\ w_i \in [u_i - \frac{1}{2}, +\infty)$ if $x_i = u_i\}$. From (4.19), the box-constrained pull-in region of x is $\mathbb{S}_x^{BC} = \{w \in \mathbb{R}^n |\ w_i \in \mathbb{B}_i,$ for $i = n, n-1, ..., 1\}$. Thus, we have

$$P_S^{BC} = P(\check{x}^{BC} = x) = P(w_i \in \mathbb{B}_i,\ i = n, n-1, ..., 1).$$

By the Chain Rule of conditional probabilities, this is equal to

$$P_S^{BC} = \prod_{i=1}^{n} P(w_i \in \mathbb{B}_i | w_n \in \mathbb{B}_n, ..., w_{i+1} \in \mathbb{B}_{i+1}).$$

However, from (3.13) we have

$$\mathrm{cov}\{w\} = \mathrm{cov}\{w - x\} = \sigma^2 D_R^{-2} = \mathrm{diag}\left(\frac{\sigma^2}{r_{11}^2}, \frac{\sigma^2}{r_{22}^2}, ..., \frac{\sigma^2}{r_{nn}^2}\right).$$

The diagonality of this VC-matrix implies that the entries of w are not correlated, therefore

$$P_S^{BC} = P(\check{x}^{BC} = x) = \prod_{i=1}^{n} P(w_i \in \mathbb{B}_i)$$

$$= \prod_{i=1}^{n} \begin{cases} \left(2\Phi\left(\frac{|r_{ii}|}{2\sigma}\right) - 1\right) & \text{if } l_i < x_i < u_i, \\ \Phi\left(\frac{|r_{ii}|}{2\sigma}\right) & \text{if } x_i = l_i \text{ or } x_i = u_i. \end{cases} \tag{5.11}$$

Remarks. By the definition of the cumulative distribution function, the inequality $2\Phi\left(\frac{|r_{ii}|}{2\sigma}\right) - 1 \leq \Phi\left(\frac{|r_{ii}|}{2\sigma}\right)$ holds for $i = 1, ..., n$. The probability that estimate $\check{x}_i^{BC}$ coincides with the true entry x_i when $l_i < x_i < u_i$ is always less than or equal to that when true $x_i = l_i$ or $x_i = u_i$. Consider how the BBNP estimator is obtained: its entries $\check{x}_i^{BC}$ are chosen in the constrained interval $[l_i, u_i]$. When $\lfloor w_i \rceil$ is not in this interval, $\check{x}_i^{BC}$ is set to either l_i or u_i, depending on which is closer to $\lfloor w_i \rceil$, otherwise $\check{x}_i^{BC} = \lfloor w_i \rceil$. Thus, the chance of $\check{x}_i^{BC}$ coinciding with x_i is higher if x_i itself is at l_i or u_i.

The difficulty that arises for box-constrained problems is that since the true integer vector x is unknown, we cannot compute the exact success rate of the BBNP estimator. However, the BBNP success rate will always lie between the following bounds:

$$\prod_{i=1}^{n}\left(2\Phi\left(\frac{|r_{ii}|}{2\sigma}\right) - 1\right) \leq P_S^{BC} \leq \prod_{i=1}^{n}\Phi\left(\frac{|r_{ii}|}{2\sigma}\right). \tag{5.12}$$

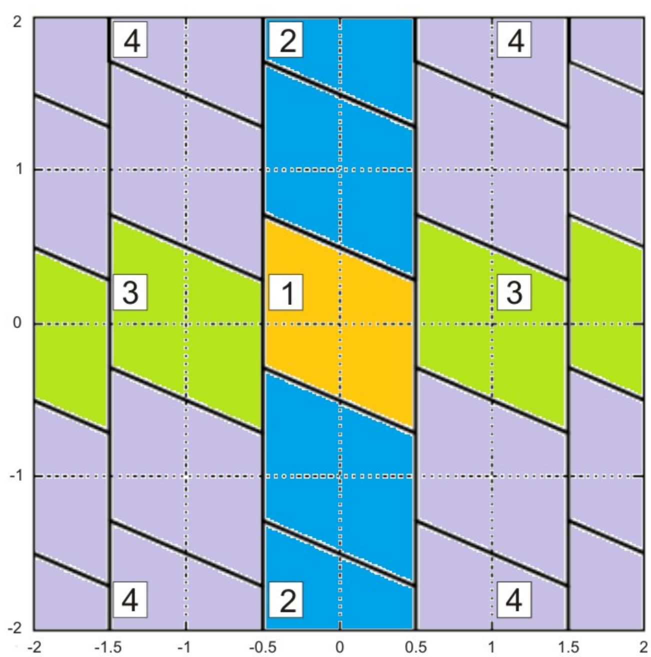

Figure 5–1: 2D BBNP success rates

2D Example. If we let box-constraints $l = [l_1, l_2]^T = [-1, -1]^T$ and $u = [u_1, u_2]^T = [1, 1]^T$, then there are four possible success rates depending on what the integer vector $x = [x_1, x_2]^T$ is equal to. Each region with the same success rate is shaded in the same color for clarity, and the four possible cases are numbered according to the cases below. For simplicity, we define $p_{1,i} = 2\Phi\left(\frac{|r_{ii}|}{2\sigma}\right) - 1$ and $p_{2,i} = \Phi\left(\frac{|r_{ii}|}{2\sigma}\right)$.

1. When $x = [0, 0]^T$, i.e. $l_1 < x_1 < u_1$ and $l_2 < x_2 < u_2$,

   the BBNP success rate is $P_S^{BC} = p_{1,1} \cdot p_{1,2}$.
2. When $x = [0, -1]^T$ or $x = [0, 1]^T$, $P_S^{BC} = p_{1,1} \cdot p_{2,2}$.
3. When $x = [-1, 0]^T$ or $x = [1, 0]^T$, $P_S^{BC} = p_{2,1} \cdot p_{1,2}$.
4. When $x = [-1, -1]^T$ or $x = [1, -1]^T$ or $x = [-1, 1]^T$ or $x = [1, 1]^T$,

   $P_S^{BC} = p_{2,1} \cdot p_{2,2}$.

Comparing the Success Rates of the BIR and BBNP Estimators. In [22], Teunissen showed that for unconstrained problems, the success rate of the BNP estimator is always greater than or equal to the success rate of the IR estimator, i.e. $P_S^B \geq P_S^R$. In box-constrained problems, this result does not hold, as it is not necessarily true that $P_S^{BC} \geq P_S^{RC}$. We give a counter example to illustrate this.

Example. Let $R = \begin{bmatrix} 2 & -1 \\ 0 & 1 \end{bmatrix}$, $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} = l$, and $\sigma = 1$. We have the following success rates:

$$P_S^{BC} = \int_{-\infty}^{1} \exp\{-\frac{1}{2}t^2\}dt \int_{-\infty}^{\frac{1}{2}} \exp\{-\frac{1}{2}t^2\}dt = \Phi\left(1\right)\Phi\left(\frac{1}{2}\right) = 0.5817583089,$$

$$P_S^{RC} = \int_{-\infty}^{\frac{1}{2}} \int_{-\infty}^{\frac{1}{2}} \frac{1}{\pi} \exp(-\frac{1}{2}\|R\xi\|^2) d\xi_2 d\xi_1 = 0.6191851371,$$

by (5.11) and (5.10) respectively. Clearly, $P_S^{BC} < P_S^{RC}$ in this example.

5.5 Success Rate of the Box-constrained Integer Least Squares Estimator

The success rate of the BILS estimator is given by

$$P_S^{IC} = P(\check{x}^{IC} = x) = \int_{\mathbb{S}_x^{IC}} f(\xi) d\xi, \tag{5.13}$$

where $\mathbb{S}_x^{IC} = \{\xi \in \mathbb{R}^n | \ \ ||\xi - x||_\Sigma^2 \leq ||\xi - \tilde{z}||_\Sigma^2, \ \forall \tilde{z} \in \mathbb{B}\}$. This probability is difficult to evaluate due to the complex integration region.

In the unconstrained case, the ILS estimator was shown to be optimal among all admissible integer estimators [23], which include the IR and BNP estimators. Furthermore, it was shown in [22] that $P_S^B \geq P_S^R$. From these results, the success rate of the BNP estimator can be used as a lower bound on the success rate of the ILS estimator, such that $P_S^I \geq P_S^B$. However, we cannot extend these results to box-constrained problems. We gave counter examples in §5.2 and §5.4 to illustrate that, due to the box-constraints, we may sometimes have $P_S^{BC} \geq P_S^{IC}$, and we may sometimes have $P_S^{RC} \geq P_S^{BC}$.

On the other hand, when the true parameter vector x is inside the box $\mathbb{B}$, i.e. $l < x < u$, the success rate probabilities of the BILS and BBNP estimators are equal to the success rate probabilities of the ILS and BNP estimators respectively, i.e. $P_S^{IC} = P_S^I$ and $P_S^{BC} = P_S^B$, since $\mathbb{S}_x^{IC} = \mathbb{S}_x^I$ and $\mathbb{S}_x^{BC} = \mathbb{S}_x^B$ for $l < x < u$. See Figures 2-2, 2-3, 4-2 and 4-3 for examples. Therefore, when $l < x < u$, we have the following lower bound on the BILS success rate:

$$P_S^{IC} = P(\check{x}^{IC} = x) \geq P(\check{x}^{BC} = x) = \prod_{i=1}^{n} \left(2\Phi\left(\frac{|r_{ii}|}{2\sigma}\right) - 1 \right). \tag{5.14}$$

Remarks on Upper Bounds. To derive an upper bound on the ILS success rate, we need the volume of the parameter search space (which is a hyper-ellipsoid), as well as the fact that the volume of any admissible pull-in region is equal to one. For details, see, e.g., [18] and [25]. The derivation itself is given in [24]. When attempting to extend this upper bound to box-constrained problems, therefore, difficulties arise due to this. To compute the volume of the box-constrained search space, we must first find the intersection of the box defined by the constraints with the ellipsoidal region, and then compute the volume of this intersected region. This is complicated. In addition, not every pull-in region has a volume equal to one. Only the regions that lie strictly inside the constraints do, while the other regions may have larger volumes. Thus we do not derive an upper bound on the BILS success rate in this thesis.

5.6 Box-constrained Partial Success Rates

The main ideas of partial validation for unconstrained problems are discussed in §3.6, and can easily be extended to box-constrained problems. In box-constrained problems, we find an integer estimate $\check{x}^C \in \mathbb{B}$ to the box-constrained integer parameter vector x using the method of BIR, BBNP or BILS. We may then compute the corresponding success rate $P_S^C = P(\check{x}^C = x)$ to evaluate the quality of the integer estimate obtained. This is the probability of a simultaneous event $\bigcap_{i=1}^{n}(\check{x}_i^C = x_i)$, and thereby it tends to decrease as n increases [28]. Therefore it may not always be possible to estimate all n entries of x with a high success rate. Given a success rate acceptability threshold P_T, the goal of partial validation is to find the largest subvector whose success rate is not smaller than P_T. When the overall success rate is less than this threshold value, i.e. $P_S^C = P(\check{x}^C = x) < P_T$, we may compute partial success rates P_{PS}^C of the entries and then fix as integers the largest subset which satisfies $P_{PS}^C \geq P_T$.

Our aim is to use partial validation in order to improve the efficiency of the BILS estimation process. The BBNP estimation method is a polynomial-time method that returns a Babai integer point solution satisfying the box constraints as an estimate of the unknown parameter vector x in (1.1). In fact, this Babai point is the first point generated during the BILS search process by the Schnorr and Euchner search. Furthermore, in theory, the BBNP success rate P_S^{BC} can be computed directly by (5.11), so we focus on the partial success rate of the BBNP estimator, denoted by P_{PS}^{BC}, in this research. Recall the BBNP success rate given in §5.4:

$$P_S^{BC} = P(\check{x}^{BC} = x) = \prod_{i=1}^{n} p_i,$$
$$\text{where } p_i = \begin{cases} 2\Phi\left(\dfrac{|r_{ii}|}{2\sigma}\right) - 1 & \text{if } l_i < x_i < u_i, \\ \Phi\left(\dfrac{|r_{ii}|}{2\sigma}\right) & \text{if } x_i = l_i \text{ or } x_i = u_i. \end{cases}$$

The probability that the subset of entries of $\check{x}^{BC}$ consisting of $\check{x}_j^{BC}$ to $\check{x}_n^{BC}$, where $i \le j \le n$, coincides with the corresponding subset of entries of true x is given by

$$\begin{aligned} P_{PS,j}^{BC} &= P([\check{x}_j^{BC}, \check{x}_{j+1}^{BC}, ..., \check{x}_n^{BC}]^T = [x_j, x_{j+1}, ..., x_n]^T) \\ &= \prod_{k=j}^{n} p_i. \end{aligned} \tag{5.15}$$

See §5.4 for details.

We can use partial validation to find the largest possible subset of the entries of the BBNP estimator $\check{x}^{BC} = [\check{x}_1^{BC}, ..., \check{x}_{n-1}^{BC}, \check{x}_n^{BC}]^T$, which has a high partial success rate, or $P_{PS}^{BC} \ge P_T$, and fix its entries as integers. We can then solve an updated but smaller BILS problem in order to find more precise integer solutions to the remaining entries of the estimate vector. Let $\tilde{x}$ denote the integer vector outcome of this partial validation process. We find $\tilde{x}$ as follows. Initially, we find $P_{PS}^{BC} = p_n$. If $P_{PS}^{BC} < P_T$, we reject the BBNP solution $\check{x}^{BC}$ and solve the BILS problem instead, to obtain $\tilde{x} = \check{x}^{IC}$. If $P_{PS}^{BC} \ge P_T$, we fix $\tilde{x}_n = \check{x}_n^{BC}$, and continue. We find $P_{PS}^{BC} = P_{PS}^{BC} \cdot p_{n-1}$, and if $P_{PS}^{BC} < P_T$, we must solve an updated BILS problem for $\tilde{x}_{n-1}$ to $\tilde{x}_1$. If $P_{PS}^{BC} \geq P_T$, we fix $\tilde{x}_{n-1} = \check{x}_{n-1}^{BC}$, and continue. Say that we find $P_{PS}^{BC} = p_n \cdot p_{n-1} \cdot \ldots \cdot p_j \geq P_T$ but $P_{PS}^{BC} = p_n \cdot p_{n-1} \cdot \ldots \cdot p_j \cdot p_{j-1} < P_T$. This means that we can successfully fix $\tilde{x}^{(2)} \equiv [\tilde{x}_j, \tilde{x}_{j+1}, ..., \tilde{x}_n]^T = [\check{x}_j^{BC}, \check{x}_{j+1}^{BC}, ..., \check{x}_n^{BC}]^T$. This is the largest possible subset of entries which have a partial success rate higher than the threshold value P_T. The remaining entries of x are computed as follows. Partition x into $\begin{bmatrix} x^{(1)} \\ \tilde{x}^{(2)} \end{bmatrix}$, where $x^{(1)} \in \mathbb{Z}^{j-1}$ and $\tilde{x}^{(2)} \in \mathbb{Z}^{n-j+1}$, with the entries of $\tilde{x}_2$ fixed to the corresponding entries of $\check{x}^{BC}$. Similarly, partition $\bar{y}$ into $\begin{bmatrix} \bar{y}^{(1)} \\ \bar{y}^{(2)} \end{bmatrix}$, $\bar{y}^{(1)} \in \mathbb{R}^{j-1}$, $\bar{y}^{(2)} \in \mathbb{R}^{n-j+1}$ and R into $\begin{bmatrix} R_1 & R_{12} \\ 0 & R_2 \end{bmatrix}$ with $R_1 \in \mathbb{R}^{(j-1)\times(j-1)}$, $R_2 \in \mathbb{R}^{(n-j+1)\times(n-j+1)}$, $R_{12} \in \mathbb{R}^{(j-1)\times(n-j+1)}$. We solve the BILS problem:

$$\min_{x^{(1)} \in \mathbb{Z}^{j-1}} \left\| \begin{bmatrix} \bar{y}^{(1)} \\ \bar{y}^{(2)} \end{bmatrix} - \begin{bmatrix} R_1 & R_{12} \\ 0 & R_2 \end{bmatrix} \begin{bmatrix} x^{(1)} \\ \tilde{x}^{(2)} \end{bmatrix} \right\|_2^2 .$$

This is equivalent to solving

$$\min_{x^{(1)} \in \mathbb{Z}^{j-1}} ||(\bar{y}^{(1)} - R_{12}\tilde{x}^{(2)}) - R_1 x^{(1)}||_2^2. \tag{5.16}$$

5.7 Numerical Simulations

All our computations were performed in MATLAB 7.9.0 (R2009b) on an AMD Phenom II X3 720 2.80GHz processor with 8GB RAM running Windows 7 Professional. In this section, we test the performance of the BBNP success rates as a measure for validating the integer estimators obtained by the BBNP method. We also compare the BBNP and BILS methods by studying the success rates of their corresponding estimators. We test the partial success rates of the BBNP estimators, and compare the partial validation method discussed in the previous section with the original BILS estimation process to see whether partial validation can improve the efficiency of BILS.

In our simulations the linear model $y = Ax + v$ of (1.1), where x is constrained to box $\mathbb{B}$ in (1.5), was constructed as follows. Without loss of generality, we took generator matrix $A \in \mathbb{R}^{n\times n}$ to be a square matrix, obtained by one of the five following cases:

- Case 1. A = randn(n,n) where randn is a built-in MATLAB function to generate an $n \times n$ matrix whose entries follow the standard normal distribution $N(0, 1)$.

Cases 2 and 3 are based on $A = L_1^T D L_2$ where L_1, L_2 are unit lower triangular matrices with entries l_{ij} $(i > j)$ generated by randn and D is generated by:

- Case 2: $D = \text{diag}(d_i)$, $d_i = \text{rand}$, where rand is a built-in MATLAB function to generate uniformly distributed random numbers in $(0, 1)$.
- Case 3: $D = \text{diag}(1^{-1}, 2^{-1}, ..., (n-1)^{-1}, n^{-1})$.

The other two cases are based on $A = UDV^T$ where U, V are random orthogonal matrices obtained by the QR factorization of two different random matrices generated by randn(n, n) and D is generated by:

- Case 4: $D = \text{diag}(d_i)$, $d_i = \text{rand}$.
- Case 5: $d_1 = 2^{-\frac{n}{4}}$, $d_n = 2^{\frac{n}{4}}$, and the other diagonal elements of D are randomly distributed between d_1 and d_n.

This allows us to test our findings on a variety of matrices with different condition numbers [7]. The condition number of square matrix A is the factor $\text{cond}(A) = ||A|| \cdot ||A^{-1}||$, and it measures the sensitivity of the solution of a system of linear equations to errors in the data. See [31] for details. The elements of the noise vector $v \in \mathbb{R}^n$ were generated randomly from $N(0, \sigma^2 I)$, and each entry of the integer parameter vector x was generated uniformly over the closed interval $[0, 3]$. The entries of the box-constraints were set to be all

0 and all 3 respectively, i.e. $l = [0, ..., 0]^T$ and $u = [3, ..., 3]^T$.

The box-constrained success rates presented in this chapter were tested as follows. For each of the five cases used to generate matrix A, we chose different σ from 0.05, 0.10, 0.25 or 0.50, and different n from 5 to 30. We tested three different measures: "true P_S^C", "theor P_S^C" and "pract P_S^C", which are obtained as follows. For each matrix A and integer parameter vector x, we generated $\mathcal{N}$ different noise vectors v according to $N(0, \sigma^2 I)$ and updated y at each run. We then solved to find $\check{x}^{BC}$ or $\check{x}^{IC}$ by either the BBNP or the BILS estimation method. We recorded the number of times $\check{x}^{BC}$ or $\check{x}^{IC}$, coincides with the true integer vector x, out of the $\mathcal{N}$ runs. This is what we refer to as the true success rate: "true P_S^{BC}" or "true P_S^{IC}" in the Tables of simulations results at the end of this section. We computed P_S^{BC}, the success rate of the BBNP estimator, by (5.11), using knowledge of true x with respect to the box-constraints l and u. In other words, we computed the success rate by checking if $l_i < x_i < u_i$ or if $x_i = l_i$ or u_i, and computing the product in (5.11) respectively. We refer to this as "theor P_S^{BC}" in the Tables of results. We also computed $\displaystyle\prod_{i=1}^{n}\left(2\Phi\left(\frac{|r_{ii}|}{2\sigma}\right) - 1\right)$, a lower bound on P_S^{BC} which can be obtained without using any knowledge of the entries of x in relation to the entries of l and u. We refer to this lower bound as "pract P_S^{BC}" in the Tables.

Tests and Results. Table 5-1 to Table 5-25 show the results obtained when testing the success rates of the BBNP estimator. We took $\sigma = 0.05, 0.10,$ $0.25, 0.50$, for $n = 5, 6, ..., 29, 30$ and $\mathcal{N} = 5000$ runs. Note that not all results are presented here. In each test, we applied both SQRD and V-BLAST reductions (see §4.4 for details) before computing the success rates and the integer estimates, in order to compare the two reduction strategies through the BBNP success rates.

When A is generated by Case 1, cond(A) is generally low and for small σ, i.e. $\sigma = 0.05$ or $\sigma = 0.10$, the success rates "true P_S^{BC}", "theor P_S^{BC}" and "pract P_S^{BC}" are close to one (see Tables 5-1 and 5-2). For $\sigma = 0.25$, the success rates are also high (see Table 5-3), so it seems that the BBNP estimator generally performs well in Case 1. More importantly, in Tables 5-1 to 5-5, we observed that "theor P_S^{BC}" is very close to the "true P_S^{BC}" from 5000 runs, thus indicating that finding the BBNP success rate (5.11) is a good approach to validating the BBNP estimator. The lower bound "pract P_S^{BC}" also seems to be a good bound in general. We also observed that, for e.g. $\sigma = 0.50$ (see Tables 5-4 and 5-5), the V-BLAST reduction strategy seems to improve the BBNP estimator, as the values of "true P_S^{BC}", "theor P_S^{BC}" and "pract P_S^{BC}" are higher than the corresponding values obtained after applying SQRD. For instance, when $n = 28$, "theor P_S^{BC}" is approximately 0.400 with SQRD, while it is approximately 0.845 with V-BLAST for the same linear model.

On the other hand, when A is generated by Case 2, cond(A) is very large (we say A is ill-conditioned) and even for small σ and small n, the success rate results are generally very low (see Table 5-6). Still, in Tables 5-6 to 5-10, we observed that "theor P_S^{BC}" is close enough to the "true P_S^{BC}" from 5000 runs, although not as close as in Case 1. We also observed that the V-BLAST reduction strategy seems to be worse than the SQRD strategy in many of the results, as the values of "true P_S^{BC}", "theor P_S^{BC}" and "pract P_S^{BC}" are lower than the corresponding values obtained after applying SQRD. For instance, when $\sigma = 0.10$ and $n = 7$ (see Table 5-8), "theor P_S^{BC}" is approximately 0.613 with SQRD, while it is approximately 0.399 with V-BLAST for the same linear model.

When A is generated by Case 3, we have similar results, as A is ill-conditioned. The success rates are close to zero as n increases for smaller

σ, or as σ increases for smaller n. See Tables 5-11 to 5-15 for details. Nevertheless, it seems "theor P_S^{BC}" is close to the "true P_S^{BC}" from 5000 runs even in this case. When A is generated by Case 4, for $\sigma = 0.05$, the success rates "true P_S^{BC}", "theor P_S^{BC}" and "pract P_S^{BC}" are high (see Table 5-16), so it seems that the BBNP estimator performs well for σ less than 0.10. The V-BLAST reduction strategy seems to generally improve the BBNP success rate results. However, as σ increases, the success rates decrease implying that the BBNP estimator degenerates for large noise (see Tables 5-19, 5-20). When A is generated by Case 5, even though cond(A) is quite large, the success rates "true P_S^{BC}", "theor P_S^{BC}" and "pract P_S^{BC}" are high (see Table 5-21 to 5-23), so it seems that the BBNP estimator performs well in Case 5. The V-BLAST reduction strategy seems to generally improve the BBNP success rate results. In all these results, we observed that the "theor P_S^{BC}" and "true P_S^{BC}" are close in value.

Table 5-26 to Table 5-30 show the results obtained when comparing the success rates of the BBNP and the BILS estimators. We took $\sigma = 0.05, 0.10, 0.25$, for $n = 5, 6, ..., 8, 9$ and $\mathcal{N} = 5000$ runs. In each test, we applied both SQRD ("SQ" in the Tables) and V-BLAST ("VB" in the Tables) reductions before computing the success rates and the integer estimates, in order to compare the two reduction strategies through the BBNP success rates. In these tests, we found "true P_S^{IC}" by computing $\check{x}^{IC}$ through Algorithm SEARCH given in [5] and then recording the number of times $\check{x}^{IC} = x$ out of the 5000 runs. We compared "theor P_S^{BC}" and "pract P_S^{BC}" to "true P_S^{IC}" to see if "theor P_S^{BC}" $\geq P_S^{IC}$ often in our results, since in §5.2 we gave a counter example showing that the optimality result of the ILS estimator (see Theorem 3.2.1) cannot be extended to the BILS estimator, and since we found that for $l < x < u$, "pract P_S^{BC}" is a lower bound to the BILS success rate (5.14). Note that we do not specifically generate x such that $l < x < u$ as we wish to see if, generally, the BBNP success rate is useful in any way for predicting the performance of the BILS estimator. The true vector x is generated uniformly over the closed interval $[0, 3]$, with the entries of l set to 0 and the entries of u set to 1.

We observed that when A is generated by Case 1, for small σ, all values are close to one (see Table 5-26). As σ increases, "true P_S^{IC}" remains higher than the corresponding BBNP results. Also, it seems that the V-BLAST reduction sharpens the lower bound "pract P_S^{BC}", making it higher than "pract P_S^{BC}" obtained after applying SQRD. However, we cannot observe this improvement in the results of "theor P_S^{BC}" and "true P_S^{BC}". When A is generated by Case 2, even for small σ, "true P_S^{IC}" is generally much higher than the corresponding BBNP success rate results. As σ increases, "true P_S^{IC}" starts to decrease (see Table 5-27). With Case 3 we observed similar results. When A is generated by Case 4, for $\sigma = 0.10$ or $\sigma = 0.25$, "true P_S^{IC}" seems to be closer to the BBNP success rate results (see Table 5-29) than in Cases 2 and 3. With Case 5, we observed high success rate results for the BBNP and the BILS estimator, in particular for $\sigma = 0.05$ and $\sigma = 0.10$.

Table 5-31 to Table 5-42 show the results obtained when testing the partial success rate of the BBNP estimator. In the Tables, "index" (j) refers to the j^{th} entry of the integer estimator $\check{x}^{BC}$, and for each entry j from $n = 20$ to 1, we compute $P_{PS,j}^{BC}$ by (5.15). This is referred to as "theor P_{PS}^{BC}" in the Tables. The "true P_{PS}^{BC}" is found by recording the number of times, out of $\mathcal{N} = 5000$ runs, that the subvector $[\check{x}_j^{BC}, \check{x}_{j+1}^{BC}, ..., \check{x}_n^{BC}]^T$ coincides with $[x_j, x_{j+1}, ..., x_n]^T$ of the true vector x. We took $\sigma = 0.10, 0.25$, for $n = 20$.

We observed, once again, that in all 5 Cases of generating matrix A, and for $\sigma = 0.10$ and $\sigma = 0.25$, the values of "theor P_{PS}^{BC}" are very close to the corresponding values of "true P_{PS}^{BC}", illustrating that the partial success rate of the BBNP estimator is a very good validation measure. We also observed that in Case 1 and Case 5, the partial success rate does not vary by much whenever another entry is considered in the product (5.15). See Tables 5-31 to 5-33, 5-41, 5-42. On the other hand, in Case 4 for example, there seems to be a fairly big drop in the value of P_{PS}^{BC} as more entries of $\check{x}^{BC}$ are considered. See Tables 5-38 to 5-40. In Cases 2 and 3, the partial success rate is low from the beginning where "index" $= j = 20$, which leads us to believe that we cannot effectively use partial validation in such cases as these, to improve the efficiency of the BILS estimation process, as none of the entries of $\check{x}^{BC}$ can be fixed, based on the results in Tables 5-34 to 5-37.

Finally, Table 5-43 to Table 5-46 show the results obtained when testing if the method of partial validation could be used to improve the efficiency of the BILS estimation process. We aim to use partial validation to fix the entries of $\check{x}^{BC}$ which have a high partial success rate, and then solve a smaller BILS problem to obtain integer estimates to the remaining entries of x. We generated A, x, l and u as described previously. We took $\sigma = 0.10, 0.50$, for $n = 5, 10, 15, 20, 25, 30$. We took the success rate acceptability threshold $P_T = 0.80$, and tried to find the largest subvector of $\check{x}^{BC}$ whose success rate is not smaller than P_T. In other words, we computed $P_{PS,j}^{BC}$ for j from n to 1, stopping when $P_{PS,j}^{BC} \geq P_T$ but $P_{PS,j-1}^{BC} < P_T$. We then set part of the integer estimate to $[\check{x}_j^{BC}, \check{x}_{j+1}^{BC}, ..., \check{x}_n^{BC}]^T$, and solved the smaller BILS problem (5.16) using Algorithm SEARCH given in [5]. See §5.6 for details. If $j = n + 1$, this means that $P_{PS,n}^{BC} < P_T$ and so we must find $\check{x}^{IC}$. If $j = -1$, this means that we can use $\check{x}^{BC}$ and we do not need to solve the BILS problem to obtain $\check{x}^{IC}$. This j is included in the Tables of results. We used the MATLAB commands tic and toc to measure the elapsed time, in seconds, of computing the partial success rate $P_{PS,j}^{BC}$ starting from n, comparing it to P_T at each step from $j = n$

to 1 (until we find $P^{BC}_{PS,j-1} < P_T$), finding $\tilde{x}$ either by the BBNP estimation method alone, or by BILS alone, or by using both methods. This measured time is referred to as "PV time" (where PV refers to partial validation) in Tables 5-43 to 5-46. We then compare this to the time taken to solve the BILS problem (4.1) without having to compute any success rates. This is referred to as "BILS time" in the Tables. For a more accurate measure of the elapsed time, we took the average of the time found by tic and toc over 100 or 5 runs (depending on the test cases). We performed similar tests using V-BLAST and SQRD reduction to test if one reduction strategy is better than the other in terms of giving higher partial success rates, which may affect the number of entries of $\check{x}^{BC}$ that can be fixed.

In Table 5-43, when A is generated by Case 1 and $\sigma = 0.10$, we observed that $j = -1$ for the different n, which means that no BILS search was needed. The "PV time" is less than the "BILS time". However, since σ is small here, even the "BILS time" is small, so the time saved by using partial validation seems to be negligible. On the other hand, when A is generated by Case 2 and $\sigma = 0.10$, we observed that $j = n + 1$ for the different n, which means that $P^{BC}_{PS,n} < P_T$ and a BILS search was required. In this case, it actually took longer time to apply partial validation, since the BILS search was carried out anyway and therefore the time taken to compute $P^{BC}_{PS,n}$ and compare it to P_T is pure overhead. We observed similar results for the other test cases as well, with the difference in "PV time" and "BILS time" generally being negligible, showing that partial validation is not helping in such cases. Similar results with respect to the time difference between "PV time" and "BILS time" were observed when SQRD reduction was applied rather than V-BLAST. We cannot effectively compare the elapsed times when V-BLAST was applied to the elapsed times when SQRD was applied as different linear models were generated to get two different sets of tests.

Tables. Table 5-1 to Table 5-46 display some results of these simulations which were carried out in order to test the box-constrained success rates found in this chapter.

Table 5–1: Testing P_S^{BC}: Case 1, $\sigma = 0.05$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	1.10e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
10	7.49e+01	0.9998	0.999048	0.999048	1.0000	1.000000	1.000000
15	9.70e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
20	1.12e+02	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
25	4.77e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
30	5.31e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000

Table 5–2: Testing P_S^{BC}: Case 1, $\sigma = 0.10$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	1.91e+01	1.0000	0.999980	0.999959	1.0000	0.999990	0.999979
6	6.44e+00	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
7	1.49e+01	0.9980	0.998127	0.996514	1.0000	0.999941	0.999941
8	2.35e+01	0.9946	0.994747	0.994747	0.9986	0.998406	0.998406
9	6.61e+02	0.5744	0.572823	0.572823	0.5744	0.572823	0.572823
10	1.36e+03	0.5436	0.553277	0.553277	0.5436	0.553277	0.553277
11	7.25e+00	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
12	6.01e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
13	1.86e+01	0.9996	0.999916	0.999832	1.0000	1.000000	1.000000
14	1.02e+02	0.9970	0.996484	0.996480	1.0000	0.999999	0.999999
15	8.45e+01	1.0000	0.999998	0.999998	1.0000	0.999998	0.999998
16	4.82e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
17	9.65e+01	1.0000	0.999999	0.999999	1.0000	1.000000	1.000000
18	1.63e+02	0.8928	0.888598	0.888598	1.0000	1.000000	1.000000
19	7.18e+01	1.0000	0.999996	0.999996	1.0000	1.000000	1.000000
20	2.08e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
21	2.53e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
22	2.75e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
23	4.04e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
24	6.05e+02	0.9896	0.991751	0.991663	0.9964	0.997109	0.994219
25	1.53e+02	0.9826	0.983484	0.983484	1.0000	1.000000	1.000000
26	5.75e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
27	3.99e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
28	1.96e+02	1.0000	0.999995	0.999995	1.0000	1.000000	1.000000
29	4.57e+02	0.6812	0.677856	0.355711	1.0000	0.999978	0.999957
30	1.97e+02	0.9506	0.950669	0.950669	1.0000	1.000000	1.000000

Table 5–3: Testing P_S^{BC}: Case 1, $\sigma = 0.25$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	2.48e+00	0.9996	0.999570	0.999141	0.9996	0.999570	0.999141
6	4.90e+00	0.9918	0.991951	0.991624	0.9918	0.991951	0.991624
7	1.57e+01	0.9924	0.993199	0.987311	0.9924	0.993199	0.987311
8	5.14e+01	0.5484	0.541491	0.541348	0.5484	0.541491	0.541348
9	8.34e+00	0.9958	0.995791	0.995552	0.9958	0.995791	0.995552
10	1.05e+01	0.9992	0.998942	0.998360	0.9988	0.998984	0.998489
11	1.62e+01	0.9900	0.990059	0.989525	0.9980	0.997505	0.996905
12	3.91e+01	0.9416	0.940622	0.940621	0.9828	0.981368	0.981353
13	5.97e+01	0.8322	0.837915	0.676759	0.9748	0.970784	0.970323
14	4.35e+01	0.9914	0.991062	0.982525	0.9816	0.982857	0.982714
15	2.18e+01	0.9914	0.991151	0.987767	0.9980	0.998599	0.998379
16	6.56e+01	0.9318	0.928596	0.889876	0.9924	0.993171	0.993008
17	3.76e+02	0.7084	0.701715	0.404316	0.8842	0.883212	0.768283
18	5.88e+01	0.9820	0.981079	0.970012	0.9954	0.995020	0.994810
19	3.16e+01	0.9860	0.986557	0.973465	0.9998	0.999802	0.999713
20	3.12e+02	0.7004	0.705655	0.412889	0.6742	0.676758	0.674286
21	1.28e+02	0.7286	0.732417	0.732406	0.9720	0.976593	0.976589
22	1.09e+02	0.7660	0.764421	0.764085	0.9380	0.939837	0.934638
23	1.09e+02	0.9860	0.986913	0.977325	0.9960	0.996901	0.995964
24	2.49e+01	0.9986	0.998806	0.998768	0.9998	0.999860	0.999843
25	1.04e+02	0.9704	0.963942	0.963937	0.9918	0.989475	0.989448
26	5.63e+02	0.3774	0.382925	0.382921	0.8706	0.872103	0.753188
27	1.41e+02	0.9740	0.973048	0.946415	0.9814	0.977597	0.975619
28	8.28e+01	0.9992	0.999156	0.998351	1.0000	0.999878	0.999798
29	2.21e+02	0.8842	0.891965	0.864650	0.9666	0.970620	0.965263
30	4.69e+01	1.0000	0.999853	0.999833	1.0000	0.999957	0.999946

Table 5–4: Testing P_S^{BC}: Case 1, $\sigma = 0.50$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	1.34e+01	0.5570	0.562522	0.558757	0.6356	0.632592	0.629905
6	3.63e+01	0.2910	0.295807	0.295807	0.2910	0.295807	0.295807
7	2.41e+01	0.5482	0.543379	0.490010	0.5772	0.567706	0.486377
8	1.74e+01	0.5804	0.579312	0.517166	0.6000	0.602210	0.520957
9	1.67e+01	0.4384	0.434929	0.421719	0.4526	0.456382	0.443188
10	2.34e+02	0.3572	0.360608	0.127384	0.3402	0.338229	0.146676
11	2.22e+01	0.3656	0.357887	0.339051	0.3862	0.386746	0.374536
12	1.86e+01	0.6590	0.651394	0.639490	0.7128	0.702333	0.682181
13	2.71e+01	0.6340	0.638729	0.574603	0.6614	0.661274	0.622051
14	9.19e+02	0.0662	0.064046	0.062362	0.1510	0.157742	0.152568
15	1.84e+01	0.7334	0.744945	0.602371	0.6900	0.691182	0.618682

Table 5–5: Case 1, $\sigma = 0.50$ (continued)

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
16	1.80e+01	0.7168	0.724260	0.717113	0.8264	0.834568	0.798712
17	2.68e+01	0.7008	0.706888	0.701318	0.7714	0.772719	0.758520
18	8.23e+01	0.3698	0.373965	0.341843	0.4986	0.493729	0.429827
19	2.66e+01	0.7864	0.783525	0.769685	0.8548	0.844831	0.830520
20	2.73e+03	0.0454	0.044923	0.027328	0.1832	0.179638	0.052817
21	5.49e+01	0.4198	0.433102	0.371794	0.5244	0.522333	0.457473
22	3.04e+01	0.8828	0.889816	0.869111	0.8932	0.895554	0.883229
23	2.80e+01	0.8740	0.869760	0.786783	0.9074	0.906422	0.867265
24	3.81e+01	0.8148	0.817324	0.801127	0.9272	0.928286	0.910652
25	4.57e+01	0.6282	0.625178	0.620988	0.7390	0.731067	0.694941
26	4.84e+01	0.8748	0.877660	0.864177	0.9080	0.903920	0.901117
27	5.03e+01	0.8242	0.824050	0.783063	0.8402	0.839989	0.808169
28	7.73e+01	0.4020	0.400059	0.395100	0.8494	0.845181	0.816128
29	1.04e+02	0.7240	0.733404	0.705868	0.7508	0.763023	0.741897
30	2.59e+01	0.9534	0.950719	0.947862	0.9662	0.965870	0.964974

Table 5–6: Testing P_S^{BC}: Case 2, $\sigma = 0.05$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	8.68e+01	0.2916	0.298549	0.298519	0.2916	0.298549	0.298519
6	1.37e+02	0.9184	0.914232	0.872547	0.9520	0.950845	0.950845
7	2.10e+02	0.7212	0.724484	0.724484	0.9646	0.966782	0.961361
8	1.65e+03	0.5492	0.551500	0.543992	0.9530	0.948855	0.920644
9	4.17e+02	0.4970	0.497915	0.162297	0.5462	0.532196	0.337299
10	2.32e+03	0.1130	0.112594	0.014391	0.0932	0.083104	0.014908
11	6.76e+03	0.0522	0.049131	0.011812	0.0932	0.084605	0.015346
12	2.34e+05	0.0000	0.000070	0.000070	0.0002	0.000610	0.000232
13	1.75e+04	0.1008	0.103933	0.036103	0.1684	0.168205	0.027296
14	2.68e+04	0.0426	0.041135	0.037421	0.3134	0.311156	0.127737
15	5.82e+04	0.0098	0.009875	0.009875	0.1300	0.130005	0.049560
16	1.54e+06	0.0170	0.017868	0.002769	0.0068	0.006329	0.003714
17	1.05e+07	0.0000	0.000094	0.000094	0.0000	0.000372	0.000256
18	8.37e+06	0.0000	0.000025	0.000025	0.0000	0.000051	0.000051
19	4.03e+07	0.0000	0.000006	0.000006	0.0002	0.000107	0.000012
20	4.95e+07	0.0002	0.000112	0.000002	0.0078	0.006733	0.000012
21	1.52e+07	0.0148	0.016890	0.000018	0.0792	0.079577	0.000090
22	4.83e+08	0.0000	0.000002	0.000002	0.0000	0.000189	0.000004
23	2.21e+06	0.0006	0.001488	0.000217	0.0004	0.000781	0.000294
24	1.65e+06	0.0000	0.000021	0.000012	0.0000	0.000022	0.000013
25	1.92e+07	0.0004	0.000330	0.000010	0.0038	0.003773	0.000043

Table 5–7: Case 2, $\sigma = 0.05$ (continued)

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
26	8.10e+06	0.0000	0.000007	0.000000	0.0000	0.000000	0.000000
27	3.90e+07	0.0000	0.000001	0.000000	0.0000	0.000112	0.000000
28	1.70e+09	0.0000	0.000000	0.000000	0.0032	0.003432	0.000000
29	3.05e+08	0.0002	0.000086	0.000000	0.0000	0.000001	0.000000
30	3.85e+08	0.0000	0.000000	0.000000	0.0092	0.010364	0.000000

Table 5–8: Testing P_S^{BC}: Case 2, $\sigma = 0.10$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	4.52e+01	0.8078	0.808161	0.670334	0.7952	0.791628	0.664404
6	3.65e+01	0.6020	0.616167	0.581517	0.6020	0.616167	0.581517
7	2.01e+03	0.6116	0.612821	0.267007	0.3916	0.398580	0.164281
8	2.12e+02	0.6346	0.645211	0.517949	0.6450	0.653103	0.557647
9	4.31e+03	0.0556	0.059339	0.012381	0.1432	0.143373	0.039314
10	5.13e+02	0.0244	0.022588	0.022467	0.0542	0.056723	0.024474
11	5.27e+04	0.0014	0.002067	0.000829	0.0034	0.003887	0.001219
12	1.02e+04	0.0448	0.045065	0.007169	0.0122	0.015412	0.008066
13	2.91e+04	0.0030	0.003212	0.003212	0.0130	0.013836	0.013836
14	4.89e+04	0.0028	0.002847	0.000100	0.0002	0.000279	0.000278
15	9.80e+04	0.0034	0.003151	0.003123	0.0430	0.040607	0.002937
16	1.91e+05	0.0000	0.000047	0.000046	0.0000	0.000133	0.000131
17	3.32e+05	0.0054	0.005365	0.000097	0.0002	0.000374	0.000182
18	3.24e+06	0.0000	0.000093	0.000040	0.0036	0.002929	0.000052
19	1.28e+07	0.0320	0.031019	0.000013	0.0222	0.023774	0.000039
20	3.45e+06	0.0004	0.000085	0.000003	0.0008	0.001420	0.000003
21	5.10e+06	0.0002	0.000237	0.000004	0.0612	0.062575	0.000003
22	3.00e+06	0.0000	0.000004	0.000001	0.0000	0.000093	0.000002
23	2.64e+06	0.0000	0.000006	0.000000	0.0000	0.000000	0.000000
24	4.15e+07	0.0000	0.000007	0.000000	0.0002	0.000073	0.000000
25	2.18e+08	0.0002	0.000527	0.000000	0.0002	0.000427	0.000001
26	2.73e+07	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
27	7.06e+07	0.0000	0.000096	0.000000	0.0000	0.000003	0.000000
28	1.19e+09	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
29	6.25e+08	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
30	4.56e+09	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000

Table 5–9: Testing P_S^{BC}: Case 2, $\sigma = 0.25$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	1.92e+02	0.1536	0.146788	0.032681	0.0288	0.027496	0.027235
6	7.04e+01	0.1948	0.188808	0.041983	0.1300	0.129048	0.031208
7	1.39e+02	0.0200	0.018945	0.012203	0.0296	0.033047	0.010828
8	2.75e+03	0.0160	0.013941	0.001050	0.0510	0.050603	0.001031
9	3.60e+03	0.0006	0.000682	0.000030	0.0000	0.000021	0.000020
10	5.86e+03	0.0002	0.000255	0.000120	0.0002	0.000275	0.000064
11	1.33e+04	0.0006	0.000374	0.000334	0.0074	0.006218	0.001318
12	9.48e+04	0.0000	0.000002	0.000000	0.0000	0.000000	0.000000
13	6.33e+03	0.0130	0.012045	0.000015	0.0000	0.000126	0.000019
14	1.21e+04	0.0052	0.004877	0.000177	0.0006	0.000211	0.000146
15	7.48e+04	0.0000	0.000348	0.000001	0.0362	0.034868	0.000008
16	5.19e+04	0.0006	0.001008	0.000001	0.0068	0.007209	0.000002
17	3.44e+04	0.0010	0.000569	0.000008	0.0024	0.001288	0.000007
18	4.04e+05	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
19	7.34e+05	0.0006	0.000372	0.000000	0.0002	0.000034	0.000000
20	1.16e+06	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
21	5.13e+08	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
22	1.82e+06	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
23	9.09e+05	0.0000	0.000024	0.000000	0.0000	0.000000	0.000000
24	1.81e+07	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
25	3.32e+07	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
26	6.72e+08	0.0000	0.000001	0.000000	0.0000	0.000001	0.000000
27	3.74e+08	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
28	6.88e+07	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
29	4.51e+09	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
30	8.64e+08	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000

Table 5–10: Testing P_S^{BC}: Case 2, $\sigma = 0.50$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	1.74e+02	0.0048	0.005194	0.000648	0.0026	0.002569	0.000635
10	2.53e+03	0.0004	0.000212	0.000091	0.0000	0.000085	0.000050
15	2.38e+04	0.0000	0.000003	0.000000	0.0000	0.000000	0.000000
20	2.37e+06	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
25	3.05e+07	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
30	1.94e+10	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000

Table 5–11: Testing P_S^{BC}: Case 3, $\sigma = 0.05$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	3.13e+01	0.9990	0.998918	0.998662	0.9990	0.998918	0.998662
6	6.11e+01	0.9162	0.913579	0.913539	0.9162	0.913579	0.913539
7	1.16e+02	0.3842	0.388325	0.388258	0.3928	0.396046	0.333390
8	1.58e+02	0.6086	0.608378	0.596273	0.6942	0.693298	0.620647
9	2.91e+02	0.6328	0.635702	0.478235	0.6150	0.612731	0.589765
10	1.38e+03	0.5850	0.586736	0.275784	0.5484	0.553813	0.370717
11	1.58e+03	0.2544	0.245573	0.180138	0.2334	0.225556	0.184844
12	8.03e+03	0.1296	0.133240	0.120382	0.1296	0.133240	0.120382
13	4.68e+03	0.2646	0.270893	0.033120	0.2694	0.272476	0.050634
14	7.25e+03	0.0652	0.072691	0.043257	0.0652	0.072691	0.043257
15	9.21e+03	0.0232	0.025088	0.014597	0.0274	0.029698	0.013326
16	6.20e+03	0.0184	0.016518	0.012293	0.0154	0.016603	0.008964
17	9.17e+03	0.0036	0.004300	0.001946	0.0026	0.002527	0.001543
18	1.47e+04	0.0222	0.020951	0.004328	0.0168	0.015804	0.004296
19	5.29e+05	0.0048	0.006403	0.000760	0.0044	0.005502	0.000664
20	1.78e+05	0.0228	0.022558	0.000254	0.0192	0.019445	0.000226
21	4.74e+06	0.0112	0.012092	0.000096	0.0096	0.010768	0.000080
22	1.41e+06	0.0002	0.000041	0.000037	0.0002	0.000041	0.000037
23	7.66e+05	0.0000	0.000346	0.000005	0.0000	0.000023	0.000005
24	6.31e+06	0.0000	0.000015	0.000002	0.0000	0.000007	0.000001
25	1.07e+06	0.0000	0.000003	0.000001	0.0000	0.000008	0.000000
26	8.22e+05	0.0002	0.000007	0.000000	0.0000	0.000006	0.000000
27	1.17e+07	0.0000	0.000001	0.000000	0.0000	0.000005	0.000000
28	1.16e+07	0.0000	0.000071	0.000000	0.0006	0.000456	0.000000
29	4.98e+07	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
30	1.12e+08	0.0000	0.000001	0.000000	0.0000	0.000001	0.000000

Table 5–12: Testing P_S^{BC}: Case 3, $\sigma = 0.10$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	2.34e+01	0.8058	0.803645	0.702696	0.8058	0.803645	0.702696
6	1.48e+02	0.4094	0.412321	0.410526	0.4094	0.412321	0.410526
7	1.12e+02	0.1938	0.201911	0.185062	0.1768	0.180254	0.177419
8	8.56e+01	0.2312	0.226452	0.143870	0.2062	0.198817	0.157482
9	1.09e+03	0.0456	0.046831	0.036673	0.0214	0.025868	0.020824
10	2.52e+03	0.1080	0.114019	0.029460	0.1036	0.111238	0.029426
11	6.54e+02	0.0084	0.009414	0.007406	0.0124	0.013905	0.006560
12	2.68e+03	0.0070	0.006898	0.002137	0.0118	0.012997	0.001895
13	2.41e+03	0.0014	0.001949	0.001104	0.0006	0.001361	0.000857
14	4.49e+03	0.0012	0.001002	0.000329	0.0012	0.001002	0.000329
15	8.12e+03	0.0004	0.000165	0.000049	0.0002	0.000157	0.000047

Table 5–13: Case 3, $\sigma = 0.10$ (continued)

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
16	7.56e+03	0.0002	0.000110	0.000018	0.0002	0.000110	0.000018
17	3.17e+04	0.0000	0.000045	0.000005	0.0000	0.000122	0.000005
18	2.22e+04	0.0000	0.000005	0.000001	0.0000	0.000004	0.000001
19	3.83e+05	0.0000	0.000005	0.000000	0.0000	0.000001	0.000000
20	5.22e+06	0.0000	0.000002	0.000000	0.0000	0.000001	0.000000
21	4.35e+05	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
22	1.98e+05	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
23	6.36e+05	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
24	5.28e+06	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
25	3.01e+05	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
26	1.06e+08	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
27	3.34e+06	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
28	3.76e+07	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
29	2.40e+08	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
30	7.21e+09	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000

Table 5–14: Testing P_S^{BC}: Case 3, $\sigma = 0.25$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	4.83e+01	0.1512	0.150818	0.052056	0.1512	0.150818	0.052056
6	5.54e+01	0.0800	0.074577	0.012236	0.0516	0.046629	0.012077
7	4.54e+02	0.0286	0.030063	0.003158	0.0286	0.030063	0.003158
8	7.78e+01	0.0008	0.001272	0.000665	0.0016	0.002101	0.000648
9	2.38e+03	0.0072	0.006316	0.000103	0.0072	0.006073	0.000103
10	1.29e+03	0.0006	0.001286	0.000016	0.0006	0.001273	0.000016
15	1.79e+04	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
20	1.47e+05	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
25	1.53e+07	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
30	3.18e+08	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000

Table 5–15: Testing P_S^{BC}: Case 3, $\sigma = 0.50$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	4.62e+01	0.0248	0.024763	0.002360	0.0248	0.024763	0.002360
10	1.31e+03	0.0000	0.000013	0.000000	0.0000	0.000013	0.000000
15	7.52e+04	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
20	1.50e+06	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
25	1.92e+07	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
30	1.10e+07	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000

Table 5–16: Testing P_S^{BC}: Case 4, $\sigma = 0.05$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	2.07e+00	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
6	8.86e+00	0.9860	0.987757	0.987756	0.9888	0.987903	0.987902
7	1.54e+01	0.9812	0.978656	0.958048	0.9818	0.978691	0.958187
8	3.36e+01	0.7446	0.739917	0.739892	0.7754	0.772000	0.771975
9	7.22e+01	0.3600	0.370090	0.189015	0.3966	0.394622	0.142467
10	1.70e+01	0.6606	0.666656	0.658501	0.6606	0.666656	0.658501
11	7.36e+01	0.3380	0.336105	0.333865	0.4722	0.477060	0.454245
12	8.87e+00	0.9948	0.995914	0.995892	0.9994	0.999595	0.999296
13	3.42e+01	0.8406	0.839540	0.838239	0.9506	0.954864	0.946625
14	3.68e+00	1.0000	0.999999	0.999999	1.0000	0.999999	0.999999
15	7.64e+01	0.7926	0.795035	0.794989	0.8442	0.848125	0.848039
16	2.31e+01	0.9720	0.971351	0.962558	0.9742	0.974928	0.973011
17	5.46e+01	0.2222	0.218643	0.210118	0.3064	0.305402	0.220064
18	1.21e+01	0.9996	0.999610	0.999608	1.0000	0.999864	0.999863
19	1.42e+01	0.9928	0.992864	0.992017	0.9922	0.992155	0.991502
20	1.46e+01	0.9856	0.985741	0.975498	0.9900	0.990815	0.987788
21	1.31e+01	0.9986	0.998671	0.998563	0.9980	0.998888	0.998666
22	3.36e+01	0.9962	0.997174	0.995736	0.9964	0.997317	0.996077
23	6.38e+01	0.9020	0.911503	0.910542	0.9928	0.993537	0.991062
24	9.15e+01	0.4286	0.426578	0.258264	0.4662	0.468565	0.404185
25	6.59e+01	0.9516	0.951014	0.942000	0.9868	0.984462	0.977138
26	2.58e+01	0.9998	0.999750	0.999724	0.9998	0.999857	0.999829
27	1.05e+01	1.0000	0.999312	0.999236	0.9998	0.999374	0.999301
28	1.23e+01	1.0000	0.999993	0.999991	1.0000	0.999994	0.999992
29	1.16e+01	0.9996	0.999387	0.999324	0.9996	0.999619	0.999607
30	3.11e+01	0.9676	0.969747	0.966673	0.9722	0.975113	0.973000

Table 5–17: Testing P_S^{BC}: Case 4, $\sigma = 0.10$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	4.20e+00	0.9372	0.937120	0.937120	0.9372	0.937120	0.937120
6	5.33e+01	0.4274	0.426823	0.342102	0.4274	0.426823	0.342102
7	9.01e+00	0.6334	0.635304	0.624753	0.6334	0.635304	0.624753
8	1.61e+01	0.7852	0.772679	0.739514	0.7852	0.772679	0.739514
9	8.96e+00	0.8204	0.817349	0.782773	0.8202	0.816648	0.785073
10	9.20e+00	0.9334	0.929558	0.920548	0.9502	0.944652	0.938717
11	2.41e+01	0.7680	0.770090	0.734707	0.7696	0.770374	0.734825
12	6.38e+00	0.7960	0.784189	0.774635	0.8344	0.825953	0.808405
13	7.61e+00	0.6768	0.671931	0.621646	0.6728	0.669971	0.618432
14	6.67e+00	0.8998	0.900267	0.845288	0.8992	0.900358	0.845463
15	1.05e+01	0.6762	0.689858	0.637488	0.6702	0.684243	0.636237

Table 5–18: Case 4, $\sigma = 0.10$ (continued)

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
16	5.36e+01	0.4904	0.495121	0.402793	0.4614	0.470574	0.403883
17	5.19e+00	0.7750	0.778180	0.708807	0.7714	0.770770	0.710979
18	6.50e+00	0.6656	0.670777	0.633703	0.6608	0.670274	0.633381
19	9.47e+00	0.5228	0.507521	0.437080	0.5202	0.510621	0.444149
20	5.02e+01	0.2566	0.258450	0.124465	0.2328	0.236304	0.200564
21	1.09e+01	0.4712	0.475890	0.417370	0.5118	0.516077	0.432627
22	6.47e+00	0.2374	0.230463	0.188747	0.2332	0.225448	0.188262
23	3.12e+01	0.0460	0.044390	0.018813	0.0468	0.047351	0.019351
24	2.24e+01	0.4196	0.413026	0.376212	0.4308	0.432508	0.384616
25	3.53e+01	0.4648	0.464900	0.419065	0.4526	0.455392	0.424120
26	9.55e+00	0.6928	0.688107	0.638489	0.6966	0.694805	0.635961
27	2.99e+01	0.3842	0.390815	0.375746	0.4322	0.431953	0.414167
28	2.89e+01	0.0300	0.031954	0.023044	0.0300	0.031526	0.023153
29	3.00e+01	0.5564	0.546011	0.490661	0.5460	0.547695	0.492828
30	1.59e+01	0.1314	0.132843	0.089540	0.1342	0.138770	0.090262

Table 5–19: Testing P_S^{BC}: Case 4, $\sigma = 0.25$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	7.89e+00	0.0660	0.066348	0.041210	0.0660	0.066348	0.041210
6	3.50e+00	0.2116	0.208844	0.048993	0.2116	0.208844	0.048993
7	2.23e+01	0.0130	0.012443	0.001346	0.0046	0.004641	0.001382
8	5.39e+00	0.0370	0.038165	0.018509	0.0264	0.030435	0.018350
9	6.12e+00	0.0408	0.040888	0.016804	0.0394	0.040315	0.016833
10	4.56e+00	0.0820	0.076850	0.053611	0.0792	0.076664	0.053528
15	4.50e+00	0.0524	0.053746	0.014563	0.0528	0.053611	0.014564
20	1.21e+01	0.0018	0.001167	0.000509	0.0016	0.001166	0.000509
25	3.14e+01	0.0000	0.000005	0.000000	0.0000	0.000008	0.000000
30	2.90e+01	0.0000	0.000002	0.000000	0.0000	0.000002	0.000000

Table 5–20: Testing P_S^{BC}: Case 4, $\sigma = 0.50$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	9.56e+00	0.0058	0.006724	0.001216	0.0042	0.006690	0.001219
10	1.36e+01	0.0000	0.000007	0.000004	0.0000	0.000009	0.000004
15	1.83e+01	0.0000	0.000003	0.000000	0.0000	0.000002	0.000000
20	2.92e+01	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
25	2.08e+01	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000
30	1.28e+01	0.0000	0.000000	0.000000	0.0000	0.000000	0.000000

Table 5–21: Testing P_S^{BC}: Case 5, $\sigma = 0.05$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	5.66e+00	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
10	3.20e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
15	1.81e+02	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
20	1.02e+03	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
25	5.79e+03	0.9996	0.999816	0.999816	1.0000	1.000000	1.000000
30	3.28e+04	1.0000	0.999998	0.999998	1.0000	1.000000	1.000000

Table 5–22: Testing P_S^{BC}: Case 5, $\sigma = 0.10$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	5.66e+00	0.9998	0.999713	0.999713	0.9998	0.999713	0.999713
6	8.00e+00	0.9998	0.999771	0.999771	0.9998	0.999771	0.999771
7	1.13e+01	0.9994	0.999620	0.999618	0.9994	0.999620	0.999618
8	1.60e+01	0.9988	0.999069	0.999069	1.0000	1.000000	1.000000
9	2.26e+01	1.0000	0.999858	0.999749	0.9998	0.999815	0.999805
10	3.20e+01	0.9690	0.965406	0.965406	0.9986	0.998035	0.998035
11	4.53e+01	1.0000	0.999986	0.999986	1.0000	1.000000	1.000000
12	6.40e+01	1.0000	0.999982	0.999982	1.0000	1.000000	1.000000
13	9.05e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
14	1.28e+02	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
15	1.81e+02	0.7000	0.694532	0.694532	1.0000	0.999982	0.999982
16	2.56e+02	1.0000	0.999999	0.999998	1.0000	1.000000	1.000000
17	3.62e+02	0.9962	0.996161	0.996161	1.0000	0.999971	0.999949
18	5.12e+02	1.0000	0.999853	0.999705	1.0000	1.000000	1.000000
19	7.24e+02	0.7140	0.715584	0.715584	0.9966	0.996907	0.996907
20	1.02e+03	0.7958	0.800463	0.600927	1.0000	0.999972	0.999972
21	1.45e+03	1.0000	0.999999	0.999999	1.0000	1.000000	1.000000
22	2.05e+03	0.9344	0.936551	0.936551	1.0000	0.999997	0.999997
23	2.90e+03	1.0000	0.999989	0.999989	1.0000	1.000000	1.000000
24	4.10e+03	0.8876	0.893528	0.893528	1.0000	1.000000	1.000000
25	5.79e+03	0.6542	0.655536	0.655536	1.0000	1.000000	1.000000
26	8.19e+03	0.6804	0.686439	0.686439	1.0000	1.000000	1.000000
27	1.16e+04	0.5522	0.555110	0.110221	0.9254	0.924663	0.854824
28	1.64e+04	0.7926	0.791691	0.583381	1.0000	1.000000	1.000000
29	2.32e+04	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
30	3.28e+04	1.0000	0.999957	0.999914	1.0000	0.999957	0.999914

Table 5–23: Testing P_S^{BC}: Case 5, $\sigma = 0.25$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	5.66e+00	0.7472	0.748770	0.748770	0.7596	0.760499	0.760499
6	8.00e+00	0.7306	0.728878	0.727328	0.7306	0.728878	0.727328
7	1.13e+01	0.8446	0.834910	0.814303	0.8178	0.813719	0.766375
8	1.60e+01	0.9648	0.964975	0.938487	0.9616	0.961770	0.956317
9	2.26e+01	0.8338	0.833982	0.833981	0.9626	0.960292	0.959901
10	3.20e+01	0.9734	0.972587	0.945177	0.9976	0.998987	0.997975
11	4.53e+01	0.7928	0.800262	0.607669	0.9038	0.906047	0.890186
12	6.40e+01	0.9888	0.988990	0.980657	0.9896	0.990470	0.981904
13	9.05e+01	1.0000	1.000000	1.000000	1.0000	1.000000	1.000000
14	1.28e+02	0.9626	0.965955	0.965955	0.9952	0.996676	0.996314
15	1.81e+02	0.9676	0.971803	0.971802	0.9986	0.998814	0.998569
16	2.56e+02	0.9888	0.989570	0.989570	0.9962	0.996828	0.996828
17	3.62e+02	0.5760	0.572820	0.572820	0.9478	0.951904	0.951904
18	5.12e+02	0.9918	0.991734	0.991734	0.9984	0.998681	0.998681
19	7.24e+02	0.7566	0.757583	0.757583	0.9984	0.998823	0.997647
20	1.02e+03	0.5510	0.550879	0.161261	0.8090	0.805806	0.753146
21	1.45e+03	0.9930	0.990457	0.990457	0.9930	0.990457	0.990457
22	2.05e+03	0.8162	0.807606	0.615212	0.6812	0.667473	0.667473
23	2.90e+03	0.8008	0.793964	0.587928	0.9772	0.976251	0.952501
24	4.10e+03	0.5634	0.561808	0.123615	0.7818	0.778989	0.557978
25	5.79e+03	0.3330	0.324495	0.324495	0.9806	0.981772	0.981772
26	8.19e+03	0.6024	0.615620	0.615620	1.0000	0.999999	0.999999
27	1.16e+04	0.6362	0.634778	0.269557	1.0000	0.999955	0.999934
28	1.64e+04	0.5836	0.584234	0.168468	0.8084	0.806102	0.612204
29	2.32e+04	0.0850	0.084694	0.084694	0.9944	0.993499	0.993499
30	3.28e+04	0.8576	0.859293	0.859293	0.9976	0.998050	0.998050

Table 5–24: Testing P_S^{BC}: Case 5, $\sigma = 0.50$

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
5	5.66e+00	0.4874	0.495229	0.464394	0.4874	0.495229	0.464394
6	8.00e+00	0.1736	0.167404	0.124077	0.1736	0.167404	0.124077
7	1.13e+01	0.1986	0.203488	0.203488	0.1954	0.200888	0.200888
8	1.60e+01	0.5468	0.551814	0.518207	0.5468	0.551814	0.518207
9	2.26e+01	0.4330	0.427574	0.381686	0.6098	0.598648	0.410104
10	3.20e+01	0.5100	0.520854	0.513496	0.6016	0.609008	0.526150
11	4.53e+01	0.2192	0.219897	0.202226	0.4556	0.444648	0.411144
12	6.40e+01	0.3692	0.374527	0.374388	0.7244	0.733675	0.731835
13	9.05e+01	0.6248	0.621561	0.621423	0.8992	0.901046	0.834522
14	1.28e+02	0.5700	0.585695	0.584131	0.5628	0.569919	0.569285
15	1.81e+02	0.4370	0.437457	0.437457	0.5932	0.594745	0.594745

Table 5–25: Case 5, $\sigma = 0.50$ (continued)

Reduction		SQRD			V-BLAST		
n	cond(A)	true P_S	theor P_S	pract P_S	true P_S	theor P_S	pract P_S
16	2.56e+02	0.2516	0.255858	0.255858	0.9742	0.974375	0.974251
17	3.62e+02	0.3826	0.378654	0.225931	0.4068	0.407704	0.367072
18	5.12e+02	0.4406	0.434340	0.386486	0.9696	0.970081	0.968691
19	7.24e+02	0.7350	0.740255	0.481208	0.8082	0.809197	0.809197
20	1.02e+03	0.3070	0.312057	0.312057	0.5186	0.531398	0.531398
21	1.45e+03	0.2598	0.255350	0.255350	0.7622	0.758144	0.758144
22	2.05e+03	0.6750	0.672773	0.351541	0.7866	0.771456	0.657535
23	2.90e+03	0.5512	0.551011	0.102022	0.5010	0.506605	0.506605
24	4.10e+03	0.8810	0.881910	0.763820	0.9316	0.930187	0.930187
25	5.79e+03	0.6704	0.661343	0.514205	0.8010	0.800956	0.793431
26	8.19e+03	0.1210	0.111173	0.111173	0.9970	0.997169	0.997157
27	1.16e+04	0.0576	0.055924	0.055924	0.6336	0.633156	0.633156
28	1.64e+04	0.0536	0.054046	0.054046	0.2578	0.257591	0.257591
29	2.32e+04	0.2830	0.285204	0.285204	0.3660	0.366091	0.366091
30	3.28e+04	0.5700	0.567655	0.135310	0.9218	0.925458	0.850921

Table 5–26: Testing P_S^{BC} and P_S^{IC}

Case	σ	n	cond(A)	Red	true P_S^{BC}	theor P_S^{BC}	pract P_S^{BC}	true P_S^{IC}
1	0.05	5	1.68e+01	SQ VB	1.0000 1.0000	1.000000 1.000000	1.000000 1.000000	1.0000
1	0.05	6	4.85e+01	SQ VB	1.0000 1.0000	0.999998 0.999998	0.999996 0.999996	1.0000
1	0.05	7	1.17e+01	SQ VB	1.0000 1.0000	1.000000 1.000000	1.000000 1.000000	1.0000
1	0.05	8	1.21e+02	SQ VB	0.8672 1.0000	0.871017 1.000000	0.742035 1.000000	1.0000
1	0.05	9	2.81e+01	SQ VB	1.0000 1.0000	1.000000 1.000000	1.000000 1.000000	1.0000
1	0.10	5	8.93e+01	SQ VB	0.9800 0.9800	0.982198 0.982198	0.964428 0.964428	1.0000
1	0.10	6	4.40e+01	SQ VB	0.9982 0.9982	0.997461 0.997461	0.997459 0.997461	1.0000
1	0.10	7	3.11e+01	SQ VB	1.0000 1.0000	1.000000 1.000000	1.000000 1.000000	1.0000
1	0.10	8	1.01e+01	SQ VB	1.0000 1.0000	1.000000 1.000000	1.000000 1.000000	1.0000
1	0.10	9	3.88e+01	SQ VB	1.0000 1.0000	0.999999 0.999999	0.999999 0.999999	1.0000

Table 5–27: Testing P_S^{BC} and P_S^{IC} (continued)

Case	σ	n	cond(A)	Red	true P_S^{BC}	theor P_S^{BC}	pract P_S^{BC}	true P_S^{IC}
1	0.25	5	2.21e+01	SQ VB	0.7952 0.6850	0.793940 0.681232	0.626595 0.662820	0.8892
1	0.25	6	8.79e+00	SQ VB	0.9630 0.9578	0.957143 0.957568	0.957143 0.957568	0.9824
1	0.25	7	2.08e+01	SQ VB	0.8782 0.9448	0.879113 0.943617	0.879113 0.943617	0.9840
1	0.25	8	4.57e+01	SQ VB	0.9718 0.9378	0.969537 0.937402	0.941182 0.937389	0.9972
1	0.25	9	1.69e+01	SQ VB	0.9184 0.9850	0.911060 0.982930	0.910899 0.974774	0.9978
2	0.05	5	3.32e+01	SQ VB	0.9928 0.9938	0.993067 0.994359	0.986762 0.989445	1.0000
2	0.05	6	1.04e+03	SQ VB	0.1918 0.5056	0.191993 0.500560	0.191993 0.461849	1.0000
2	0.05	7	2.10e+03	SQ VB	0.1212 0.1430	0.118179 0.138507	0.111341 0.130493	0.9428
2	0.05	8	1.63e+03	SQ VB	0.0878 0.4392	0.088117 0.445787	0.088117 0.445787	0.9872
2	0.05	9	7.80e+02	SQ VB	0.1858 0.2134	0.176643 0.207863	0.037939 0.122965	0.9988
2	0.10	5	2.13e+02	SQ VB	0.3764 0.2010	0.369741 0.193567	0.050094 0.069613	0.9978
2	0.10	6	1.81e+03	SQ VB	0.5132 0.0638	0.513781 0.067821	0.041559 0.049177	0.9946
2	0.10	7	3.11e+03	SQ VB	0.0134 0.0124	0.011709 0.011696	0.011709 0.011696	0.9376
2	0.10	8	2.53e+03	SQ VB	0.4634 0.4634	0.462587 0.462587	0.206540 0.206540	0.9998
2	0.10	9	3.64e+02	SQ VB	0.1594 0.3784	0.163025 0.385667	0.163025 0.385667	1.0000
2	0.25	5	1.31e+02	SQ VB	0.0040 0.0040	0.005008 0.005008	0.003374 0.003374	0.0130
2	0.25	6	1.63e+03	SQ VB	0.0180 0.0178	0.020772 0.020483	0.009793 0.009533	0.7872
2	0.25	7	2.57e+02	SQ VB	0.0108 0.0336	0.008976 0.033074	0.007526 0.007558	0.3700
2	0.25	8	2.59e+02	SQ VB	0.0132 0.0428	0.013292 0.041441	0.012097 0.012313	0.4884
2	0.25	9	5.02e+03	SQ VB	0.0096 0.0006	0.008038 0.000601	0.000165 0.000221	0.3516

Table 5–28: Testing P_S^{BC} and P_S^{IC} (continued)

Case	σ	n	cond(A)	Red	true P_S^{BC}	theor P_S^{BC}	pract P_S^{BC}	true P_S^{IC}
3	0.05	5	1.29e+01	SQ VB	0.9944 0.9944	0.994585 0.994584	0.994524 0.994523	0.9966
3	0.05	6	2.35e+02	SQ VB	0.9290 0.9530	0.930308 0.952547	0.930308 0.952547	0.9772
3	0.05	7	1.36e+02	SQ VB	0.8734 0.8734	0.875646 0.875646	0.858106 0.858106	1.0000
3	0.05	8	2.15e+02	SQ VB	0.7052 0.7018	0.706130 0.701675	0.696588 0.695695	0.9960
3	0.05	9	3.60e+03	SQ VB	0.1404 0.1260	0.142873 0.123684	0.138014 0.122844	0.9980
3	0.10	5	3.18e+01	SQ VB	0.5788 0.7408	0.592081 0.748216	0.544703 0.581470	0.8890
3	0.10	6	8.93e+01	SQ VB	0.4500 0.4500	0.447419 0.447419	0.369374 0.369374	0.8064
3	0.10	7	2.43e+02	SQ VB	0.2632 0.2632	0.259893 0.259893	0.251958 0.251958	0.8828
3	0.10	8	1.62e+02	SQ VB	0.2910 0.2844	0.283379 0.275989	0.164676 0.160381	0.5364
3	0.10	9	4.35e+02	SQ VB	0.1654 0.1790	0.164607 0.179616	0.039699 0.042228	0.6586
3	0.25	5	4.53e+01	SQ VB	0.1008 0.0892	0.104164 0.092559	0.046244 0.044919	0.5360
3	0.25	6	1.64e+01	SQ VB	0.0314 0.0314	0.031015 0.031015	0.014119 0.014119	0.0432
3	0.25	7	1.13e+02	SQ VB	0.0278 0.0256	0.030114 0.027488	0.003234 0.003258	0.1646
3	0.25	8	1.47e+01	SQ VB	0.0008 0.0014	0.001196 0.001340	0.000705 0.000712	0.0038
3	0.25	9	6.39e+02	SQ VB	0.0004 0.0004	0.000469 0.000469	0.000125 0.000125	0.0154
4	0.05	5	2.83e+00	SQ VB	1.0000 1.0000	1.000000 1.000000	1.000000 1.000000	1.0000
4	0.05	6	4.98e+00	SQ VB	0.9986 0.9986	0.999421 0.999421	0.999232 0.999232	0.9996
4	0.05	7	2.53e+00	SQ VB	1.0000 1.0000	1.000000 1.000000	1.000000 1.000000	1.0000
4	0.05	8	5.15e+00	SQ VB	0.9998 1.0000	0.999965 0.999949	0.999932 0.999945	1.0000
4	0.05	9	4.07e+00	SQ VB	1.0000 1.0000	0.999956 0.999956	0.999947 0.999947	1.0000

Table 5–29: Testing P_S^{BC} and P_S^{IC} (continued)

Case	σ	n	cond(A)	Red	true P_S^{BC}	theor P_S^{BC}	pract P_S^{BC}	true P_S^{IC}
4	0.10	5	2.31e+01	SQ VB	0.1308 0.1204	0.122560 0.115184	0.051629 0.053182	0.2270
4	0.10	6	4.05e+00	SQ VB	0.7366 0.7696	0.745636 0.770252	0.733712 0.736285	0.7854
4	0.10	7	1.08e+01	SQ VB	0.4226 0.4226	0.413936 0.413936	0.305742 0.305742	0.5194
4	0.10	8	9.79e+00	SQ VB	0.6352 0.6352	0.645671 0.645671	0.640747 0.640747	0.8450
4	0.10	9	5.81e+00	SQ VB	0.7546 0.7492	0.753872 0.749772	0.740870 0.740228	0.8830
4	0.25	5	1.15e+01	SQ VB	0.0094 0.0094	0.011524 0.011524	0.011524 0.011524	0.0194
4	0.25	6	6.93e+01	SQ VB	0.0102 0.0102	0.012689 0.012689	0.005293 0.005293	0.3054
4	0.25	7	4.32e+01	SQ VB	0.0048 0.0064	0.004967 0.005658	0.000407 0.000407	0.0718
4	0.25	8	1.81e+01	SQ VB	0.0058 0.0070	0.007025 0.006261	0.007025 0.006261	0.0298
4	0.25	9	8.31e+00	SQ VB	0.0064 0.0070	0.005069 0.005163	0.001542 0.001542	0.0102
5	0.05	5	5.66e+00	SQ VB	1.0000 1.0000	1.000000 1.000000	1.000000 1.000000	1.0000
5	0.05	6	8.00e+00	SQ VB	1.0000 1.0000	1.000000 1.000000	1.000000 1.000000	1.0000
5	0.05	7	1.13e+01	SQ VB	1.0000 1.0000	1.000000 1.000000	1.000000 1.000000	1.0000
5	0.05	8	1.60e+01	SQ VB	1.0000 1.0000	1.000000 1.000000	1.000000 1.000000	1.0000
5	0.05	9	2.26e+01	SQ VB	1.0000 1.0000	1.000000 1.000000	1.000000 1.000000	1.0000
5	0.10	5	5.66e+00	SQRD VB	1.0000 1.0000	0.999992 0.999992	0.999985 0.999985	1.0000
5	0.10	6	8.00e+00	SQ VB	1.0000 1.0000	0.999900 0.999900	0.999900 0.999900	1.0000
5	0.10	7	1.13e+01	SQ VB	0.9966 1.0000	0.997672 0.999947	0.997672 0.999947	1.0000
5	0.10	8	1.60e+01	SQ VB	1.0000 1.0000	0.999998 0.999998	0.999998 0.999998	1.0000
5	0.10	9	2.26e+01	SQ VB	0.9954 1.0000	0.995318 0.999937	0.990636 0.999873	1.0000

Table 5–30: Testing P_S^{BC} and P_S^{IC} (continued)

Case	σ	n	cond(A)	Red	true P_S^{BC}	theor P_S^{BC}	pract P_S^{BC}	true P_S^{IC}
5	0.25	5	5.66e+00	SQ	0.6526	0.650185	0.592070	0.7054
				VB	0.6526	0.650185	0.592070	
5	0.25	6	8.00e+00	SQ	0.9048	0.901362	0.896310	0.9478
				VB	0.9048	0.901362	0.896310	
5	0.25	7	1.13e+01	SQ	0.9810	0.982629	0.980163	0.9956
				VB	0.9810	0.982629	0.980163	
5	0.25	8	1.60e+01	SQ	0.9232	0.920710	0.841604	1.0000
				VB	0.9978	0.997239	0.996233	
5	0.25	9	2.26e+01	SQ	0.7116	0.709369	0.660841	0.8546
				VB	0.6932	0.697386	0.649678	

Table 5–31: Testing P_{PS}^{BC}: Case 1, $\sigma = 0.10$, cond(A) = 5.32e+01

Red	SQRD		V-BLAST	
index	true P_{PS}^{BC}	theor P_{PS}^{BC}	true P_{PS}^{BC}	theor P_{PS}^{BC}
20	1.0000	1.0000000000	1.0000	1.0000000000
19	1.0000	1.0000000000	1.0000	1.0000000000
18	1.0000	0.9999999992	1.0000	1.0000000000
17	1.0000	0.9999999990	1.0000	1.0000000000
16	1.0000	0.9999999990	1.0000	1.0000000000
15	1.0000	0.9999999990	1.0000	1.0000000000
10	1.0000	0.9999999990	1.0000	1.0000000000
5	1.0000	0.9999999990	1.0000	1.0000000000
1	1.0000	0.9999999990	1.0000	1.0000000000

Table 5–32: Testing P_{PS}^{BC}: Case 1, $\sigma = 0.25$, cond(A) = 2.44e+01

Red	SQRD		V-BLAST	
index	true P_{PS}^{BC}	theor P_{PS}^{BC}	true P_{PS}^{BC}	theor P_{PS}^{BC}
20	1.0000	0.9999997622	1.0000	0.9999997622
19	1.0000	0.9999919471	1.0000	0.9999919471
18	1.0000	0.9999862129	1.0000	0.9999862129
17	1.0000	0.9999748379	1.0000	0.9999820033
16	1.0000	0.9998597613	1.0000	0.9999814819
15	1.0000	0.9998597333	1.0000	0.9999789312
14	1.0000	0.9998596121	1.0000	0.9999685658
13	1.0000	0.9998591400	1.0000	0.9999626208
12	1.0000	0.9998591339	1.0000	0.9999594189
11	1.0000	0.9998591326	1.0000	0.9999561301
10	1.0000	0.9998591119	1.0000	0.9999561019

Table 5–33: Case 1, $\sigma = 0.25$, cond(A) = 2.44e+01 (continued)

Red	SQRD		V-BLAST	
index	true P_{PS}^{BC}	theor P_{PS}^{BC}	true P_{PS}^{BC}	theor P_{PS}^{BC}
9	1.0000	0.9998591091	1.0000	0.9999561013
8	1.0000	0.9998591080	1.0000	0.9999561012
7	1.0000	0.9998585984	1.0000	0.9999560982
6	1.0000	0.9998585979	1.0000	0.9999560978
5	1.0000	0.9998585979	1.0000	0.9999560968
4	1.0000	0.9998585979	1.0000	0.9999560968
3	1.0000	0.9998585970	1.0000	0.9999560968
2	1.0000	0.9998585970	1.0000	0.9999560968
1	1.0000	0.9998585969	1.0000	0.9999560967

Table 5–34: Testing P_{PS}^{BC}: Case 2, $\sigma = 0.10$, cond(A) = 6.72e+05

Red	SQRD		V-BLAST	
index	true P_{PS}^{BC}	theor P_{PS}^{BC}	true P_{PS}^{BC}	theor P_{PS}^{BC}
20	0.0394	0.0347659978	0.0394	0.0347659978
19	0.0000	0.0000279063	0.0214	0.0176629198
18	0.0000	0.0000162026	0.0038	0.0044392926
17	0.0000	0.0000060389	0.0012	0.0011225871
16	0.0000	0.0000060289	0.0006	0.0007847015
15	0.0000	0.0000060213	0.0004	0.0005860718
14	0.0000	0.0000060213	0.0004	0.0005860718
13	0.0000	0.0000060067	0.0004	0.0005860718
12	0.0000	0.0000060067	0.0004	0.0005859834
11	0.0000	0.0000060051	0.0004	0.0005857958
10	0.0000	0.0000060051	0.0004	0.0005857958
9	0.0000	0.0000060051	0.0004	0.0005857958
8	0.0000	0.0000060051	0.0004	0.0005857958
7	0.0000	0.0000055278	0.0004	0.0005392395
6	0.0000	0.0000055278	0.0004	0.0005392395
5	0.0000	0.0000028399	0.0004	0.0005390534
4	0.0000	0.0000028399	0.0004	0.0005378073
3	0.0000	0.0000028399	0.0004	0.0005313526
2	0.0000	0.0000028399	0.0004	0.0005313526
1	0.0000	0.0000028399	0.0004	0.0005313526

Table 5–35: Testing P_{PS}^{BC}: Case 2, $\sigma = 0.25$, cond(A) = 4.36e+06

Red	SQRD		V-BLAST	
index	true P_{PS}^{BC}	theor P_{PS}^{BC}	true P_{PS}^{BC}	theor P_{PS}^{BC}
20	0.0012	0.0011729953	0.0012	0.0011729953
19	0.0000	0.0000087609	0.0000	0.0000102496
18	0.0000	0.0000057800	0.0000	0.0000021612
17	0.0000	0.0000016409	0.0000	0.0000014781
16	0.0000	0.0000007832	0.0000	0.0000007055
15	0.0000	0.0000004563	0.0000	0.0000004110
14	0.0000	0.0000004371	0.0000	0.0000003937
13	0.0000	0.0000003365	0.0000	0.0000003031
12	0.0000	0.0000000411	0.0000	0.0000002330
11	0.0000	0.0000000311	0.0000	0.0000001424
10	0.0000	0.0000000296	0.0000	0.0000000521
9	0.0000	0.0000000286	0.0000	0.0000000455
8	0.0000	0.0000000246	0.0000	0.0000000455
7	0.0000	0.0000000223	0.0000	0.0000000455
6	0.0000	0.0000000223	0.0000	0.0000000424
5	0.0000	0.0000000223	0.0000	0.0000000424
1	0.0000	0.0000000223	0.0000	0.0000000424

Table 5–36: Testing P_{PS}^{BC}: Case 3, $\sigma = 0.10$, cond(A) = 1.12e+05

Red	SQRD		V-BLAST	
index	true P_{PS}^{BC}	theor P_{PS}^{BC}	true P_{PS}^{BC}	theor P_{PS}^{BC}
20	0.0412	0.0390428341	0.0412	0.0390428341
19	0.0276	0.0256882268	0.0276	0.0256882268
18	0.0026	0.0025496961	0.0034	0.0027681289
17	0.0010	0.0009872550	0.0022	0.0018895751
16	0.0010	0.0005830963	0.0002	0.0003338426
15	0.0008	0.0003538300	0.0000	0.0001382319
14	0.0002	0.0000950513	0.0000	0.0001063779
13	0.0002	0.0000593970	0.0000	0.0000334818
12	0.0000	0.0000459035	0.0000	0.0000268870
11	0.0000	0.0000383183	0.0000	0.0000133902
10	0.0000	0.0000321528	0.0000	0.0000094915
9	0.0000	0.0000262554	0.0000	0.0000066762
8	0.0000	0.0000198439	0.0000	0.0000057162
7	0.0000	0.0000132564	0.0000	0.0000043915
6	0.0000	0.0000089750	0.0000	0.0000034191
5	0.0000	0.0000056961	0.0000	0.0000021700
4	0.0000	0.0000036903	0.0000	0.0000014059
3	0.0000	0.0000023050	0.0000	0.0000008781
2	0.0000	0.0000016225	0.0000	0.0000006181
1	0.0000	0.0000011215	0.0000	0.0000004272

Table 5–37: Testing P_{PS}^{BC}: Case 3, $\sigma = 0.25$, cond(A) = 3.80e+04

Red	SQRD		V-BLAST	
index	true P_{PS}^{BC}	theor P_{PS}^{BC}	true P_{PS}^{BC}	theor P_{PS}^{BC}
20	0.0106	0.0115982849	0.0106	0.0115982849
19	0.0002	0.0001902891	0.0052	0.0061361408
18	0.0000	0.0001135186	0.0028	0.0031622679
17	0.0000	0.0000715025	0.0008	0.0003891998
16	0.0000	0.0000062061	0.0002	0.0002597755
15	0.0000	0.0000016344	0.0000	0.0000614324
14	0.0000	0.0000009430	0.0000	0.0000161786
13	0.0000	0.0000005690	0.0000	0.0000095935
12	0.0000	0.0000001791	0.0000	0.0000058316
11	0.0000	0.0000001155	0.0000	0.0000018441
10	0.0000	0.0000000283	0.0000	0.0000011896
9	0.0000	0.0000000103	0.0000	0.0000002941
8	0.0000	0.0000000069	0.0000	0.0000001143
7	0.0000	0.0000000044	0.0000	0.0000000739
6	0.0000	0.0000000012	0.0000	0.0000000196
5	0.0000	0.0000000007	0.0000	0.0000000124
4	0.0000	0.0000000002	0.0000	0.0000000033
3	0.0000	0.0000000001	0.0000	0.0000000009
2	0.0000	0.0000000000	0.0000	0.0000000006
1	0.0000	0.0000000000	0.0000	0.0000000002

Table 5–38: Testing P_{PS}^{BC}: Case 4, $\sigma = 0.10$, cond(A) = 3.46e+01

Red	SQRD		V-BLAST	
index	true P_{PS}^{BC}	theor P_{PS}^{BC}	true P_{PS}^{BC}	theor P_{PS}^{BC}
20	0.6384	0.6402729318	0.8728	0.8669283600
19	0.5362	0.5380359588	0.5574	0.5568362352
18	0.4592	0.4618373620	0.4702	0.4699938982
17	0.3236	0.3207814346	0.4064	0.4058696977
16	0.2490	0.2468222577	0.3022	0.2973078950
15	0.2264	0.2233662878	0.2274	0.2203527176
14	0.2088	0.2026114439	0.1642	0.1591214934
13	0.1592	0.1552988095	0.1352	0.1298702230
12	0.1230	0.1209137921	0.1284	0.1225204543
11	0.0884	0.0891829161	0.1010	0.0984671933
10	0.0758	0.0752539525	0.0842	0.0808210495
9	0.0732	0.0709131933	0.0794	0.0766770235
8	0.0698	0.0650474292	0.0716	0.0687125922
7	0.0642	0.0598143560	0.0630	0.0611369879
6	0.0604	0.0561444024	0.0538	0.0526175896
5	0.0556	0.0513618848	0.0516	0.0494086003

Table 5–39: Case 4, $\sigma = 0.10$, cond(A) = 3.46e+01 (continued)

Red	SQRD		V-BLAST	
index	true P_{PS}^{BC}	theor P_{PS}^{BC}	true P_{PS}^{BC}	theor P_{PS}^{BC}
4	0.0530	0.0495213279	0.0512	0.0487962775
3	0.0516	0.0477980211	0.0498	0.0473372497
2	0.0508	0.0470760245	0.0486	0.0460589733
1	0.0488	0.0443324618	0.0478	0.0433746836

Table 5–40: Testing P_{PS}^{BC}: Case 4, $\sigma = 0.25$, cond(A) = 8.29e+00

Red	SQRD		V-BLAST	
index	true P_{PS}^{BC}	theor P_{PS}^{BC}	true P_{PS}^{BC}	theor P_{PS}^{BC}
20	0.7846	0.7884627990	0.7846	0.7884627990
19	0.5426	0.5559414152	0.6778	0.6832245886
18	0.4766	0.4839950546	0.4774	0.4835594502
17	0.4116	0.4205050618	0.4186	0.4214498839
16	0.3090	0.3067072842	0.3134	0.3073964171
15	0.2250	0.2185349268	0.2274	0.2190259474
14	0.1506	0.1439331066	0.1518	0.1442565062
13	0.1266	0.1215797521	0.1284	0.1218529266
12	0.1080	0.1041501288	0.1088	0.1043841411
11	0.0894	0.0884803465	0.0896	0.0886791507
10	0.0652	0.0594023380	0.0618	0.0622139183
9	0.0564	0.0508332667	0.0516	0.0522681206
8	0.0404	0.0375569962	0.0388	0.0386707992
7	0.0340	0.0321249051	0.0330	0.0330776121
6	0.0244	0.0229662097	0.0234	0.0236473033
5	0.0192	0.0167993271	0.0184	0.0172975336
4	0.0132	0.0120184349	0.0140	0.0123748577
3	0.0104	0.0087573628	0.0106	0.0090170741
2	0.0086	0.0075864447	0.0086	0.0078114308
1	0.0068	0.0055397569	0.0068	0.0057040458

Table 5–41: Testing P_{PS}^{BC}: Case 5, $\sigma = 0.10$, cond(A) = 1.02e+03

Red	SQRD		V-BLAST	
index	true P_{PS}^{BC}	theor P_{PS}^{BC}	true P_{PS}^{BC}	theor P_{PS}^{BC}
20	1.0000	1.0000000000	1.0000	1.0000000000
19	0.9784	0.9755612100	1.0000	0.9999418768
18	0.9784	0.9755612100	1.0000	0.9999418768
17	0.9784	0.9755612100	1.0000	0.9999418768
16	0.9784	0.9755612100	1.0000	0.9999418768
15	0.9784	0.9755612100	1.0000	0.9999418768
10	0.9784	0.9755612100	1.0000	0.9999418768
5	0.9784	0.9755612100	1.0000	0.9999418768
1	0.9784	0.9755612100	1.0000	0.9999418768

Table 5–42: Testing P_{PS}^{BC}: Case 5, $\sigma = 0.25$, cond(A) = 1.02e+03

Red	SQRD		V-BLAST	
index	true P_{PS}^{BC}	theor P_{PS}^{BC}	true P_{PS}^{BC}	theor P_{PS}^{BC}
20	0.5096	0.5155922694	0.7820	0.7799789954
19	0.5092	0.5147497709	0.7742	0.7700900797
18	0.5092	0.5147497709	0.7742	0.7700792617
17	0.5092	0.5147486653	0.7742	0.7700792617
16	0.5092	0.5147486653	0.7742	0.7700792617
15	0.5092	0.5147193225	0.7742	0.7700792617
14	0.5092	0.5147193225	0.7742	0.7700792617
13	0.5092	0.5147193225	0.7742	0.7700792617
12	0.5092	0.5147193225	0.7742	0.7700792617
11	0.5092	0.5147193225	0.7742	0.7700792617
10	0.5092	0.5147193225	0.7742	0.7700792617
5	0.5092	0.5147193225	0.7742	0.7700792617
1	0.5092	0.5147193225	0.7742	0.7700792617

Table 5–43: Testing Partial Validation, with V-BLAST

Case	σ	cond(A)	n	j	PV time (seconds)	BILS time (seconds)
1	0.10	1.68e+01	5	-1	0.000299	0.000320
1	0.10	6.92e+02	10	-1	0.000294	0.000343
1	0.10	1.18e+02	15	-1	0.000428	0.000511
1	0.10	2.53e+01	20	-1	0.000572	0.000693
1	0.10	4.08e+01	25	-1	0.000710	0.000863
1	0.10	4.83e+01	30	-1	0.000841	0.001049
1	0.50	1.53e+01	5	5	0.000250	0.000200
1	0.50	1.14e+01	10	6	0.000427	0.000597
1	0.50	1.13e+02	15	12	0.002586	0.003069
1	0.50	5.12e+01	20	21	0.001288	0.001099
1	0.50	3.15e+02	25	25	0.045641	0.045786
1	0.50	2.41e+02	30	31	0.576652	0.575157
2	0.10	1.52e+02	5	6	0.001126	0.000739
2	0.10	4.98e+02	10	11	0.002784	0.002675
2	0.10	8.87e+04	15	16	0.423797	0.425264
2	0.10	7.73e+05	20	21	3.512154	3.511518
2	0.10	4.19e+07	25	26	2736.456531	2737.545857
2	0.50	1.81e+02	5	6	0.000469	0.000399
2	0.50	8.66e+03	10	11	0.030135	0.029923
2	0.50	4.77e+05	15	16	21.879887	21.870041
2	0.50	3.18e+06	20	21	19.914085	19.927906
2	0.50	1.08e+07	25	26	3552.010936	3552.527851

Table 5–44: Testing Partial Validation, with V-BLAST (continued)

Case	σ	cond(A)	n	j	PV time (seconds)	BILS time (seconds)
3	0.10	1.14e+02	5	5	0.000752	0.000597
3	0.10	3.12e+03	10	11	0.001258	0.001153
3	0.10	4.90e+04	15	16	0.195215	0.195138
3	0.10	5.22e+04	20	21	31.952989	31.949389
3	0.10	5.46e+05	25	26	55.823419	55.841507
3	0.10	2.58e+07	30	31	33037.186392	33036.531852
3	0.50	2.50e+01	5	6	0.000495	0.000426
3	0.50	4.59e+02	10	11	0.008374	0.008214
3	0.50	1.02e+04	15	16	0.173584	0.171770
3	0.50	7.21e+04	20	21	509.934067	509.903094
3	0.50	1.39e+06	25	26	1172.297615	1171.655575
3	0.50	1.84e+07	30	31	7526.823317	7525.267589
4	0.10	1.75e+01	5	5	0.000527	0.000260
4	0.10	4.18e+00	10	-1	0.000290	0.000341
4	0.10	4.34e+01	15	16	0.011302	0.011135
4	0.10	7.72e+01	20	21	0.008473	0.008285
4	0.10	2.52e+01	25	24	0.091388	0.104592
4	0.10	6.87e+01	30	31	0.937605	0.937842
4	0.50	4.06e+00	5	6	0.000241	0.000172
4	0.50	2.90e+01	10	11	0.013848	0.013736
4	0.50	7.91e+00	15	16	0.143757	0.143433
4	0.50	9.37e+01	20	21	0.044110	0.043811
4	0.50	1.59e+01	25	26	4.070929	4.096489
4	0.50	7.63e+01	30	31	53.046567	53.004471
5	0.10	5.66e+00	5	-1	0.000148	0.000171
5	0.10	3.20e+01	10	-1	0.000288	0.000340
5	0.10	1.81e+02	15	-1	0.000425	0.000511
5	0.10	1.02e+03	20	-1	0.000562	0.000686
5	0.10	5.79e+03	25	-1	0.000714	0.000884
5	0.10	3.28e+04	30	-1	0.000867	0.001044
5	0.50	5.66e+00	5	4	0.000237	0.000175
5	0.50	3.20e+01	10	10	0.001282	0.001222
5	0.50	1.81e+02	15	16	0.014658	0.014445
5	0.50	1.02e+03	20	-1	0.000565	0.000736
5	0.50	5.79e+03	25	-1	0.000705	0.000859
5	0.50	3.28e+04	30	30	0.001399	0.001129

Table 5–45: Testing Partial Validation, with SQRD

Case	σ	cond(A)	n	j	PV time (seconds)	BILS time (seconds)
1	0.10	3.70e+01	5	-1	0.000150	0.000175
1	0.10	2.30e+02	10	11	0.001207	0.001108
1	0.10	2.70e+02	15	-1	0.000428	0.000512
1	0.10	2.45e+01	20	-1	0.000575	0.000684
1	0.10	6.87e+01	25	-1	0.000714	0.000867
1	0.10	7.81e+02	30	-1	0.000847	0.001087
1	0.50	1.98e+01	5	6	0.000523	0.000453
1	0.50	2.20e+01	10	10	0.000759	0.000987
1	0.50	3.43e+01	15	11	0.000600	0.000513
1	0.50	5.83e+02	20	21	7.493023	7.494661
1	0.50	5.88e+01	25	-1	0.000717	0.001143
1	0.50	4.27e+01	30	-1	0.000857	0.003543
2	0.10	3.73e+01	5	4	0.000266	0.000204
2	0.10	9.16e+03	10	11	0.008671	0.008532
2	0.10	1.04e+05	15	16	0.176747	0.175916
2	0.10	3.09e+07	20	21	4.094464	4.090744
2	0.10	2.09e+08	25	26	10.157302	10.146382
2	0.50	1.81e+02	5	6	0.000499	0.000428
2	0.50	3.23e+03	10	11	0.022505	0.022255
2	0.50	3.95e+05	15	16	0.102387	0.102209
2	0.50	5.58e+05	20	21	23.992044	24.006116
2	0.50	1.01e+10	25	26	32.696803	32.732611
3	0.10	7.20e+01	5	4	0.000266	0.000285
3	0.10	2.76e+02	10	11	0.014830	0.014662
3	0.10	9.96e+03	15	16	2.306317	2.310466
3	0.10	6.56e+05	20	21	38.304451	38.274589
3	0.10	1.33e+07	25	26	2622.926651	2621.117036
3	0.10	1.24e+07	30	31	223.104373	223.197020
3	0.50	6.45e+00	5	6	0.000835	0.000773
3	0.50	2.35e+03	10	11	0.032682	0.032497
3	0.50	6.68e+03	15	16	0.630040	0.630224
3	0.50	4.88e+04	20	21	3.524793	3.521879
3	0.50	9.73e+07	25	26	8400.725470	8397.624178
3	0.50	2.58e+07	30	31	19869.786378	19850.928084

Table 5–46: Testing Partial Validation, with SQRD (continued)

Case	σ	cond(A)	n	j	PV time (seconds)	BILS time (seconds)
4	0.10	3.41e+01	5	5	0.000609	0.000230
4	0.10	5.19e+00	10	4	0.000380	0.000448
4	0.10	7.59e+01	15	16	0.031082	0.030810
4	0.10	9.32e+00	20	16	0.001072	0.001303
4	0.10	1.32e+01	25	24	0.008568	0.012787
4	0.10	7.39e+00	30	24	0.001213	0.002446
4	0.50	2.64e+01	5	6	0.000496	0.000429
4	0.50	1.74e+01	10	11	0.002808	0.002714
4	0.50	1.14e+01	15	16	0.026375	0.026122
4	0.50	1.86e+01	20	21	9.525524	9.524562
4	0.50	3.17e+01	25	26	0.422697	0.421923
4	0.50	4.36e+01	30	31	0.125745	0.125163
5	0.10	5.66e+00	5	-1	0.000152	0.000173
5	0.10	3.20e+01	10	-1	0.000287	0.000342
5	0.10	1.81e+02	15	-1	0.000429	0.000569
5	0.10	1.02e+03	20	21	0.035586	0.035202
5	0.10	5.79e+03	25	-1	0.000711	0.000866
5	0.10	3.28e+04	30	-1	0.000851	0.001042
5	0.50	5.66e+00	5	4	0.000235	0.000203
5	0.50	3.20e+01	10	-1	0.000291	0.000341
5	0.50	1.81e+02	15	-1	0.000430	0.000593
5	0.50	1.02e+03	20	20	0.046896	0.046231
5	0.50	5.79e+03	25	26	0.001172	0.000959
5	0.50	3.28e+04	30	31	0.515869	0.516052

CHAPTER 6
Summary and Future Work

In this thesis, we extended the theory of success rates to box-constrained linear models of the form:

$$y = Ax + v, \ v \sim N(0, \sigma^2 I),$$

where x is the unknown integer parameter vector to be estimated, and it is subject to the box-constraint $\mathbb{B} = \{x \in \mathbb{Z}^n | \ l \leq x \leq u, \ l \in \mathbb{Z}^n, u \in \mathbb{Z}^n\}$.

In chapter 2, we reviewed the typical methods for estimating $x \in \mathbb{Z}^n$, i.e. in the unconstrained case, namely the integer rounding (IR), Babai nearest plane (BNP) and integer least squares (ILS) estimation methods. The most effective approach to validating an integer estimator is to find its success rate, which is the probability of correct integer estimation. In chapter 3, we reviewed the success rates of the (unconstrained) IR, BNP and ILS estimators and presented some of their properties which are given in GPS literature. One such property is that the ILS estimator is optimal among all admissible integer estimators, including the IR and BNP estimators. In chapter 4, we presented the box-constrained integer rounding (BIR), box-constrained Babai nearest plane (BBNP) and box-constrained integer least squares (BILS) methods of estimating $x \in \mathbb{B}$. In chapter 5, we extended the theory of success rates to the BIR, BBNP and BILS estimators. In particular, we gave examples to show that some properties of success rates, such as the optimality of the ILS estimator, which hold in unconstrained problems, do not hold in box-constrained problems. Furthermore, we extended the theory of partial validation to box-constrained estimators, to investigate whether or not it could be applied to improve the efficiency of the BILS estimation process. In particular, if a subvector of the Babai point estimate obtained by BBNP has a high partial success rate, we wanted to see if we could fix these entries and then solve a smaller BILS problem to get integer estimates to the remaining entries of x.

Numerical simulations results showed that even for box-constrained problems, the success rate of the BBNP estimator is a good measure of validating the BBNP estimator. However, numerical simulations results also showed that partial validation cannot as yet be effectively used to improve the efficiency of the BILS estimation process. We observed only small differences in the time taken to estimate x through partial validation and the time taken to estimate x through BILS.

In the future, we would like to develop sharp lower bounds on the success rate of the BILS estimator, which is equal to $P_S^{IC} = \int\limits_{\mathbb{S}_x^{IC}} f(\xi)d(\xi)$, by bounding this region of integration $\mathbb{S}_x^{IC}$. We would also like to further explore the possibility of improving the efficiency of the BILS estimation process using the idea of partial validation. In [5], a new reduction strategy was proposed and shown to be more effective than the V-BLAST and SQRD reduction strategies that we considered in this research. Furthermore, the Babai point obtained by this reduction strategy is usually closer to the ILS solution than the Babai point obtained by V-BLAST or SQRD [5]. However, this reduction uses the input vector y as well as $\mathbb{B}$. We would like to extend our success rates to be able to efficiently validate the integer parameter estimator obtained through this reduction strategy. We could then use partial success rates to fix more entries of the integer estimate thus obtained, and as this will further reduce the size of the BILS problem to be solved, perhaps it will give larger differences in the computational time taken for applying the partial validation method and the time taken to solve the original BILS problem.

REFERENCES

[1] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory*, 48(8):2201–2214, 2002.

[2] L. Babai. On Lovász's lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[3] P. van Emde Boas. Another NP-complete partition problem and the complexity of computing short vectors in a lattice. Technical Report 81-04, Mathematisch Institute, Amsterdam, The Netherlands, 1981.

[4] J. Boutros, N. Gresset, L. Brunel, and M. Fossorier. Soft-input soft-output lattice sphere decoder for linear channels. In *Proceedings of IEEE 2003 Global Communications Conference (GLOBECOM 2003)*, pages 213–217, San Francisco, California, USA, December 1-5 2003.

[5] X.-W. Chang and Q. Han. Solving box-constrained integer least-squares problems. *IEEE Transactions on Wireless Communications*, 7(1):277–287, 2008.

[6] X.-W. Chang and C.C. Paige. Euclidean distances and least squares problems for a given set of vectors. *Applied Numerical Mathematics*, 57:1240–1244, 2007.

[7] X.-W. Chang, X. Yang, and T. Zhou. MLAMBDA: a modified LAMBDA method for integer least-squares estimation. *Journal of Geodesy*, 79:552–565, 2005.

[8] X.-W. Chang and T. Zhou. *MILES: MATLAB package for solving Mixed Integer LEast Squares problems (V1.1)*, 2011.

[9] M.O. Damen, H. El Gamal, and G. Caire. On maximum-likelihood detection and the search for the closest lattice point. *IEEE Transactions on Information Theory*, 49(10):2389–2402, 2003.

[10] G.J. Foschini, G.D. Golden, R.A. Valenzuela, and P.W. Wolniansky. Simplified processing for high spectral efficiency wireless communication employing multi-element arrays. *IEEE Journal on Selected Areas in Communications*, 17(11):1841–1852, 1999.

[11] A. Hassibi and S. Boyd. Integer parameter estimation in linear models with applications to GPS. *IEEE Transactions on Signal Processing*, 46(11):2938–2952, 1998.

[12] N.L. Johnson, S. Kotz, and N. Balakrishnan. Chi-square distributions including Chi and Raleigh. In *Continuous Univariate Distributions*, pages 415–493. John Wiley and Sons, New York, 2nd edition, 1994.

[13] A.K. Lenstra, J.H.W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[14] C. Ling and N. Howgrave-Graham. Effective LLL reduction for lattice decoding. In *IEEE International Symposium on Information Theory (ISIT 2007)*, pages 196–200, Nice, France, June 24-29 2007.

[15] D. Micciancio. The hardness of the closest vector problem with preprocessing. *IEEE Transactions on Information Theory*, 47:1212–1215, 2001.

[16] W.H. Mow. Universal lattice decoding: principle and recent advances. *Wireless Communications and Mobile Computing*, 3(5):553–569, 2003.

[17] C.P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66:181–199, 1994.

[18] P.J.G. Teunissen. A canonical theory for short GPS baselines, part iv: precision versus reliability. *Journal of Geodesy*, 71(9):513–525, 1997.

[19] P.J.G. Teunissen. A class of unbiased integer GPS ambiguity estimators. *Artificial Satellites*, 33(1):3–10, 1998.

[20] P.J.G. Teunissen. GPS carrier phase ambiguity fixing concepts. In *GPS for Geodesy*, pages 317–388. Springer-Verlag, New York, 2nd edition, 1998.

[21] P.J.G. Teunissen. On the integer normal distribution of the GPS ambiguities. *Artificial Satellites*, 33(2):49–64, 1998.

[22] P.J.G. Teunissen. Success probability of integer GPS ambiguity rounding and bootstrapping. *Journal of Geodesy*, 72(10):606–612, 1998.

[23] P.J.G. Teunissen. An optimality property of the integer least squares estimator. *Journal of Geodesy*, 73(11):587–593, 1999.

[24] P.J.G. Teunissen. ADOP-based upperbounds for the bootstrapped and the least-squares ambiguity success rates. *Artificial Satellites*, 35(4):171–179, 2000.

[25] P.J.G. Teunissen. The parameter distributions of the integer GPS model. *Journal of Geodesy*, 76(1):41–48, 2002.

[26] P.J.G. Teunissen. Penalized GNSS ambiguity resolution. *Journal of Geodesy*, 78(4-5):235–244, 2004.

[27] P.J.G. Teunissen and D. Odijk. Ambiguity Dilution of Precision: definition, properties and applications. In *Proceedings of the 10th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1997)*, pages 891–899, Kansas City, Missouri, USA, September 16-19 1997.

[28] P.J.G. Teunissen and S. Verhagen. GNSS carrier phase ambiguity resolution: challenges and open problems. In *Observing our Changing Earth*, volume 133 of *International Association of Geodesy Symposia*, pages 785–792, Perugia, Italy, July 2-13 2007.

[29] M. Thomson and P. Schmidt. A note on the comparison of the mean square error of inequality constrained least squares and other related estimators. *The Review of Economics and Statistics*, 64(1):174–176, 1982.

[30] S. Verhagen. *The GNSS integer ambiguities: estimation and validation*. PhD thesis, TU Delft, 2005.

[31] D.S. Watkins. Sensitivity of linear systems; effects of roundoff errors. In *Fundamentals of Matrix Computations*, pages 111–180. Wiley-Interscience, New York, 2nd edition, 2002.

[32] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K.D. Kammeyer. Efficient algorithm for decoding layered space-time codes. *IEEE Electronics Letters*, 37(22):1348–1350, 2001.

[33] G. Xu. In *GPS: Theory, Algorithms and Applications*. Springer, Berlin, 2nd edition, 2007.

[34] P. Xu. Voronoi cells, probabilistic bounds, and hypothesis testing in mixed integer linear models. *IEEE Transactions on Information Theory*, 52(7):3122–3138, 2006.

[35] S. Zhang, H. Bian, and Q. Zhang. Monitoring large-area mine deformation by GPS considering plate movement. In *International Conference on Consumer Electronics, Communications and Networks (CECNet 2011)*, pages 5060–5063, XianNing, China, April 16-18 2011.

[36] T. Zhou. Modified LLL algorithms. Master's thesis, McGill University, 2006.