

Development of Equipment to Characterize Soil Attributes in Different Agricultural Settings

Eko Budi Bowo Leksono

Faculty of Agricultural and Environmental Sciences

Department of Bioresource Engineering

Macdonald Campus of McGill University

21,111 Lakeshore Road, Ste. Anne de Bellevue, Québec H9X 3V9 Montreal, Canada

May 2020

A Thesis submitted to McGill University in partial fulfillment of the requirements of the degree of
Doctor of Philosophy

© Eko Budi Bowo Leksono, Canada, 2020

Dedication

This dissertation is dedicated to my son, my inspiration, Muhammad Aqeel Fattan who teaches me how to be happy, patient, and alive. Importantly, to my home country Indonesia where due to its natural wealth “a wooden stick and stone can be a plant”. I hope my dissertation brings a positive impact on the precision agriculture development in Indonesia.

Abstract

Agriculture intensification of crop production through heavy and uniform fertilization has harmful consequences. Precision agriculture (PA) offers a methodology to mitigate the negative effects of heavy fertilization by monitoring available soil nutrients to prevent excess fertilizing. Variable rate fertilizer application (VRA) provides information about the existing soil nutrients and their location in the field, thus, allowing farmers to apply fertilizer effectively. Georeferenced soil chemical information, such as soil pH and soluble nitrate, is important for successful VRA recommendation maps. This information can be measured using ion-selective electrode (ISE) sensors directly *in-situ*. Spot measurement using an ISE can be done by making firm contact with the soil surface, known as direct soil measurement (DSM). Reliable automatic and manual soil sensing platforms are needed to perform DSM using ISEs. In this research, an automatic vehicle mounted on-the-spot soil analyzer (OSA) was developed. The OSA works by digging the soil to the depth defined by the operator, conduct DSM using multiple ISEs and cover the sampling hole and wash the ISEs when finished measuring. The OSA digging controller was able to manage the digging load accordingly under various field conditions. The OSA operation took 60 s per location to complete. Both the pH and nitrate ISEs were able to satisfactorily predict soil pH and nitrate with R^2 of 0.59 and 0.72, respectively with RMSE of 0.43 pH and 0.16 pNO₃. As an OSA alternative, a portable manual soil sensing platform was developed to accommodate the needs of small scale farmers and to provide a general overview of specific soil properties in the field. The manual soil sensing platform developed in this research offers a simple operation. Similar to standard soil core sampling, the operator pushes the platform into the soil. Then, conducting DSM by bringing the ISEs into contact with the sampled soil. The operator can monitor the ISE's reading from the Arduino based DAQ connected to a generic Android Bluetooth terminal application. Manual pumping action will spray water to clean the ISEs after the measuring is finished. The developed manual soil analyzer provides georeferenced ISEs data and it is able to satisfactorily predict soil pH and nitrate with R^2 between 0.53 to 0.88 with RMSE ranging from 0.17 to 0.36 pH for soil pH and R^2 between 0.83 and 0.84 with RMSE of 0.21 to 0.29 pNO₃ for soil nitrate prediction. Apparent soil electrical conductivity (EC_a) maps provide valuable information for determining suitable sampling locations for an

effective soil sensing platform operation. The interpretation of the EC_a measurements is often site-specific, thus, needing an inversion to properly delineate the EC_a data from the electromagnetic induction (EMI) sensor. In general, there are two approaches to invert the EMI measurements into depth specific soil EC_a information: the finite element and fixed slice cumulative depth response approach. In this research, Brute-Force cumulative depth response inversion was developed due to its simplicity and straight forward calculation utilizing the low induction number (LIN) assumption of the EMI sensor. From filtered DUALEM-21S (Duaem, Inc. Milton, Ontario, Canada) EC_a data, the Brute-Force EC_a inversion produced two layer soil maps with their corresponding EC_a and depth information. The software was tested successfully for characterizing the depth of the muck soil layer in a typical horticulture farming operation in Quebec, Canada.

Résumé

L'intensification de la production agricole par une fertilisation lourde et uniforme a des conséquences néfastes. L'agriculture de précision (AP) offre une méthodologie pour atténuer les effets négatifs de la fertilisation intensive en surveillant les nutriments disponibles dans le sol pour éviter une fertilisation excessive. L'application d'engrais à taux variable (VRA) fournit des informations sur les éléments nutritifs du sol existants et leur emplacement dans le champ, permettant ainsi aux agriculteurs d'appliquer efficacement l'engrais. Les informations chimiques géoréférencées du sol, telles que le pH du sol et le nitrate soluble, sont importantes pour la réussite des cartes de recommandation VRA. Ces informations peuvent être mesurées à l'aide de capteurs à électrode sélective ionique (ISE) directement in situ. La mesure ponctuelle à l'aide de l'ISE peut être effectuée en établissant un contact ferme avec la surface du sol, connue sous le nom de mesure directe du sol (DSM). Des plates-formes de détection du sol automatiques et manuelles fiables sont nécessaires pour effectuer le DSM à l'aide d'ISE. Dans cette recherche, un analyseur automatique de sol monté sur place (OSA) a été développé. L'OSA fonctionne en creusant le sol à la profondeur définie par l'opérateur, en effectuant le DSM en utilisant plusieurs ISE et en couvrant le trou d'échantillonnage et en lavant les ISE une fois la mesure terminée. Le contrôleur d'excavation OSA a pu gérer la charge d'excavation en conséquence dans diverses conditions de terrain. L'opération OSA a pris 60 s par emplacement pour se terminer. Les ISE de pH et de nitrate ont pu prédire de manière satisfaisante le pH et le nitrate du sol avec R^2 de 0,59 et 0,72, respectivement avec RMSE de 0,43 pH et 0,16 pNO₃. Comme alternative à l'OSA, une plate-forme portable de détection manuelle des sols a été développée pour répondre aux besoins des petits agriculteurs et pour fournir un aperçu général des propriétés spécifiques des sols sur le terrain. La plateforme de détection manuelle du sol développée dans cette recherche offre une opération simple. Semblable à l'échantillonnage de carottes de sol standard, l'opérateur pousse la plate-forme dans le sol. Ensuite, effectuer le DSM en mettant les ISE en contact avec le sol échantillonné. L'opérateur peut surveiller la lecture de l'ISE à partir du DAQ basé sur Arduino connecté à une application de terminal Bluetooth Android générique. L'action de pompage manuel

pulvériser de l'eau pour nettoyer les ISE une fois la mesure terminée. L'analyseur de sol manuel développé fournit des données ISE géoréférencées et il est capable de prédire de manière satisfaisante le pH du sol et le nitrate avec R^2 entre 0,53 à 0,88 avec RMSE allant de 0,17 à 0,36 pH pour le pH du sol et R^2 entre 0,83 et 0,84 avec RMSE de 0,21 à 0,29 pNO₃ pour la prévision du nitrate du sol. Les cartes de conductivité électrique apparente (EC_a) du sol fournissent des informations précieuses pour déterminer les emplacements d'échantillonnage appropriés pour un fonctionnement efficace de la plateforme de détection du sol. L'interprétation des mesures EC_a est souvent spécifique au site, donc, nécessitant une inversion pour délimiter correctement les données EC_a du capteur d'induction électromagnétique (EMI). En général, il existe deux approches pour inverser les mesures EMI en informations EC_a du sol spécifique à la profondeur: l'approche par réponse cumulative en profondeur des éléments finis et des tranches fixes. Dans cette recherche, l'inversion de réponse en profondeur cumulée Brute-Force a été développée en raison de sa simplicité et de son calcul simple utilisant l'hypothèse de faible nombre d'induction (LIN) du capteur EMI. À partir des données EC_a DUALEM-21S filtrées (Dualém, Inc. Milton, Ontario, Canada), l'inversion Brute-Force EC_a a produit des cartes de sol à deux couches avec leurs informations EC_a et profondeur correspondante. Le logiciel a été testé avec succès pour caractériser la profondeur de la couche de terre noire sur une exploitation horticole typique au Québec, au Canada.

Acknowledgements

I would like to acknowledge Indonesia Endowment Fund for Education (LPDP) and the National Science and Engineering Research Council (NSERC) Discovery Grant for supporting me financially during my research.

I would also like to thank Professor Viacheslav I. Adamchuk as my thesis supervisor, who taught me how to handle pressure, converting problems into life enjoyment and providing countless life advice.

I thank Professor Joann Whalen and Professor Abdolhamid Akbarzadeh Shafaroudi for their wisdom, expertise, and valuable input for my research. I would also like to thank Helene Lalonde, Hicham Benslim, Dr. Michael Boh, Scott Manktelow, and Guillaume Cloutier (Delfland Farm) for helping me with the laboratory work, prototype making and field testing.

I acknowledge the contributions of the past and current Precision Agriculture and Sensor Systems (PASS) research team members Dr. Hsin-Hui Huang, Dr. Park Jae Sung, Dr. Wenjun Ji, Sophie Lauzon, Maxime Leclerc, Roberto Buelvas, Md. Saifuzzaman, Pierce Carlson, Trevor Stanhope, Florian Reumont, Evan Henry, Yue Su, and many others for helping me with my research.

I thank Dr. Bezgin, Dr. Suarhana, Bapak Endro family, Syiar Montreal, PPI Montreal and Indonesian family here in Canada for their immense help and support.

Contributions of Author

List of publications and scientific presentations related to the thesis:

A. Thesis chapters submitted for publication in peered reviewed journals

1. **Leksono, E.**, V. Adamchuk, M. Leclerc, R. Buelvas, C. Miller, J. Park, and P. Carlson. Development of an on-the-spot automated soil analyzer for subsurface measurement of soil properties. *Computers and Electronics in Agriculture* (to be submitted).

Author Contributions: Conceptualization, Viacheslav Adamchuk, Eko Leksono and Maxime Leclerc; methodology, Eko Leksono, Maxime Leclerc and Connor Miller; software, Eko Leksono and Roberto Buelvas; formal analysis, Eko Leksono; investigation, Eko Leksono, Jaesung Park and Pierce Carlson; resources, Eko Leksono; visualization, Eko Leksono.

2. **Leksono, E.**, V.I. Adamchuk, J. Whalen, M. Leclerc, and J. Park. Development of a manual soil analyzer for measuring multiple chemical soil properties in the field. *Sensors* (to be submitted).

Author Contributions: Conceptualization, Viacheslav Adamchuk and Eko Leksono; methodology, Eko Leksono; software, Eko Leksono; formal analysis, Eko Leksono; investigation, Eko Leksono, Joann Whalen, Maxime Leclerc and Jaesung Park; resources, Eko Leksono and Joann Whalen; visualization, Eko Leksono.

3. **Leksono, E.**, V. Adamchuk, M. Leclerc, W. Ji, and A. Akbarzadeh. Development and validation of Brute-Force soil EC_a inversion for characterizing the top organic soil layer. *Journal of Environmental and Engineering Geophysics* (to be submitted).

Author Contributions: Conceptualization, Viacheslav Adamchuk, Eko Leksono and Abdolhamid Akbarzadeh; methodology, Eko Leksono and Viacheslav Adamchuk; software, Eko Leksono; formal analysis, Eko Leksono; investigation, Eko Leksono, Maxime Leclerc and Wenjun Ji; resources, Eko Leksono; visualization, Eko Leksono.

B. Non-refereed conference proceedings

1. **Leksono, E.** and V.I. Adamchuk. 2016. Development of a portable multiple ion-selective electrodes apparatus for rapid soil nitrate measurement. In: Proceedings of the 1st International Conference on the Role of Agricultural Engineering of Sustainable Agriculture Production (AESAP), Bogor, Indonesia, 13-14 December 2016,153-164. Bogor, Indonesia: Bogor Agricultural University.
2. **Leksono, E.**, V. Adamchuk, W. Ji, and M. Leclerc. 2018. Development of a soil EC_a inversion algorithm for topsoil depth characterization. In: Proceedings of the 14th International Conference on Precision Agriculture, Montreal, Quebec, Canada. 24-27 June 2018. International Society of Precision Agriculture (published on-line at <http://www.ispag.org>, 11 pages).
3. **Leksono, E.**, V. Adamchuk, H. Akbarzadeh, M. Leclerc, and R. Buelvas. 2018. Development of a subsoil manual electrical conductivity probe. Paper No. 18-201. Orleans, Ontario: CSBE.
4. **Leksono, E.**, V. Adamchuk, M. Leclerc, R. Buelvas, C. Miller, and J. Park. 2019. Development of an on-the-spot proximal soil sensing platform for subsurface measurement of soil properties. In: Proceedings of the 5th Global Workshop on Proximal Soil Sensing, 27-31 May 2019, Columbia, Missouri, USA, eds. K.A. Sudduth, N.R. Kitchen, and K.S. Veum, 205-210. Columbia, Missouri, USA: USDA-ARS.

C. Poster presentation

Leksono, E., V. Adamchuk, J. Whalen, and R. Buelvas. 2018. Development of a manual soil sensing system for measuring multiple chemical soil properties in the field. In: Proceedings of the 14th International Conference on Precision Agriculture, Montreal, Quebec, Canada. 24-27 June 2018. International Society of Precision Agriculture (published on-line at <http://www.ispag.org>, 5 pages)

Table of Contents

Dedication.....	ii
Abstract.....	iii
Résumé.....	v
Acknowledgements.....	vii
Contributions of Author.....	viii
A. Thesis chapters submitted for publication in peered reviewed journals	viii
B. Non-refereed conference proceedings	ix
C. Poster presentation	ix
Table of Contents.....	x
List of Tables.....	xiii
List of Figures	xiv
Abbreviation	xvii
1. Introduction.....	1
1.1. General Introduction	1
1.2. Statement of Rationale	3
1.3. Objectives of the Research.....	3
2. General Review of the Literature	4
2.1. Soil Chemical Investigation.....	4
2.2. Soil Sensing Platform.....	7
2.2.1. Automatic Soil Sensing Platform.....	7
2.2.2. Manual Soil Sensing Platform	14
2.3. Soil EC _a Inversion.....	19
Connection to Chapter 3	26
3. Development of an Automatic Soil Analyzer.....	27
Abstract.....	27
3.1. Introduction.....	28
3.2. Materials and Methods	29
3.2.1. OSA Operation	29
3.2.2. Physical Design	30
3.2.3. Electronics Components	31
3.2.4. OSA Data Acquisition System (DAQ) and Graphical User Interface (GUI)	34
3.2.5. OSA Control Systems.....	35

3.2.6.	OSA Field Operation.....	62
3.2.7.	OSA Field Test	63
3.3.	Results and Discussion	65
3.3.1.	OSA Controller Performance.....	65
3.3.2.	OSA Direct Soil Measurement Assessment.....	67
3.4.	Conclusion.....	69
	Connection to Chapter 4	70
4.	Development of Manual Soil Analyzer	71
	Abstract.....	71
4.1.	Introduction.....	72
4.2.	Materials and Methods	74
4.2.1.	Development of Soil Sampler	74
4.2.2.	Electronics and Data Acquisition System (DAQ).....	75
4.2.3.	Field Operation	78
4.2.4.	Field Test.....	79
4.3.	Results and Discussion	80
4.3.1.	Tentacle Noise	80
4.3.2.	Tentacle Cross Talk and Ground Loop.....	82
4.3.3.	Manual Soil Analyzer Field Operation.....	84
4.3.4.	DAQ Performance	84
4.4.	Conclusion.....	87
	Connection to Chapter 5	88
5.	Development of Brute-Force Soil EC _a Inversion Algorithm	89
	Abstract.....	89
5.1.	Introduction.....	90
5.2.	Materials and Methods	92
5.2.1.	DUALEM-21S Mapping	92
5.2.2.	DUALEM-21S Data Filtering.....	92
5.2.3.	DUALEM-21S Response Function.....	96
5.2.4.	Brute-Force Inversion Software	98
5.2.5.	Development of a Manual EC _a Probe	102
5.2.6.	Soil Layering Model from Subsoil Electrical Conductivity Probe Data.....	105
5.3.	Results and Discussion	106

5.3.1.	Brute-Force Inversion Outputs	106
5.3.2.	EC _a Inversion Maps	107
5.3.3.	Soil EC _a Model to Determine the Depth of Muck Soil Layer.....	111
5.3.4.	Depth of Muck Soil Layer Estimation	116
5.4.	Conclusion.....	119
6.	Summary and General Conclusion.....	120
6.1.	Summary.....	120
6.2.	General Conclusion.....	122
7.	Contributions to Knowledge and Suggestions for Future Research	123
7.1.	Contribution to Knowledge.....	123
7.2.	Suggestions for Future Research	124
8.	References	126
9.	Appendices.....	136
9.1.	Automatic Soil Analyzer Electronic Diagrams.....	136
9.2.	Automatic Soil Analyzer Control and DAQ Code	139
9.2.1.	Digging Controller Code.....	139
9.2.2.	ISEs Deployment Controller Code.....	148
9.2.3.	OSA DAQ Code.....	152
9.3.	PD Compensation Matlab Code.....	158
9.4.	Manual Soil Sampler DAQ Code	160
9.5.	DUALEM-21S Default NMEA 0183 Data Stream.....	165
9.6.	DUALEM-21S Filtering Code.....	166
9.7.	Brute-Force EC _a Inversion Code.....	176

List of Tables

Table 3-1. Ziegler-Nichols rules and result	52
Table 3-2. Comparison between uncompensated and PD compensated system	55
Table 3-3. Control design parameter summary	56
Table 4-1. Tentacle channel 1 crosstalk test sequence	77
Table 4-2. ISEs used in the crosstalk test	77
Table 4-3. Manual soil analyzer ISEs performance	84
Table 5-1. Depth of muck soil estimation from EC _a model and Brute-Force inversion	117

List of Figures

Figure 2-1. ISE’s potentiometry circuit..... 7

Figure 2-2. Tractor mounted on-the-go soil measurement..... 8

Figure 2-3. Schematic of the on-the-go platform. 9

Figure 2-4. Agitated soil chamber module (ACM) arrangements.10

Figure 2-5. On-the-go soil pH and lime requirement measuring system.11

Figure 2-6. On-the-go soil sensing platform based on near infrared spectrophotometer.....11

Figure 2-7. On-the-go soil sensing platform based on visible-near infrared sensors.....12

Figure 2-8. AgriRover: an autonomous soil nitrate sensing platform.....13

Figure 2-9. On-the-spot soil analyzer.14

Figure 2-10. Multi ISFETs flow injection analysis apparatus.....15

Figure 2-11. Soil core nitrate extraction system and analyzer using ISFETs.16

Figure 2-12. ISEs based benchtop soil analyzer.16

Figure 2-13. Benchtop in-situ soil nitrate analyzer using ISEs with soil extraction system.17

Figure 2-14. LIBS based in-situ soil analyzer.18

Figure 2-15. Manual probe for on-the-spot soil measurement.19

Figure 2-16. Typical electromagnetic induction (EMI) sensor working principle20

Figure 2-17. Finite element EMI inversion flowchart.....22

Figure 2-18. Fixed slice cumulative depth response EMI inversion flowchart22

Figure 2-19. Difference in soil EC model (A) and inversion result (B).....24

Figure 3-1. The OSA operation flow29

Figure 3-2. The new OSA design30

Figure 3-3. OSA digging curvature comparison.....31

Figure 3-4. The detailed OSA electronical layout33

Figure 3-5. LabVIEW OSA GUI.....34

Figure 3-6. The OSA digging block diagram.....36

Figure 3-7. The OSA digging depth block diagram36

Figure 3-8. The OSA flatten blades block diagram37

Figure 3-9. The OSA sensor deployment block diagram37

Figure 3-10. The OSA washing block diagram38

Figure 3-11. The OSA soil covering block diagram38

Figure 3-12. Frame linear actuator model response40

Figure 3-13. Linear actuator Simulink model42

Figure 3-14. Frame linear actuator PWM motor controller response43

Figure 3-15. Blade motor simulation response after adjustment	44
Figure 3-16. The angle between linear actuator and blade frame when digging	46
Figure 3-17. White-box OSA digging controller model.....	47
Figure 3-18. Comparison of 5 points average and median filter	48
Figure 3-19. OSA blade current response	49
Figure 3-20. OSA plant second-order transfer function identification GUI.....	50
Figure 3-21. S-shaped curve controller's field test response	52
Figure 3-22. Step response PID controller	53
Figure 3-23. Step response PD controller	54
Figure 3-24. Uncompensated step response.....	55
Figure 3-25. PD compensated step response	55
Figure 3-26. OSA controller test at grass field	57
Figure 3-27. Controller output of the OSA control system during field validation test.....	58
Figure 3-28. Digging current of the OSA control system during field validation test.....	59
Figure 3-29. Digging depth of the OSA control system during field validation test	59
Figure 3-30. Controller output of the OSA control system during field tuning.....	60
Figure 3-31. Digging current of the OSA control system during field tuning	61
Figure 3-32. Digging depth of the OSA control system during field tuning	61
Figure 3-33. Typical field tuning digging result	62
Figure 3-34. The OSA field setup	62
Figure 3-35. The OSA field operation.....	63
Figure 3-36. OSA digging control output in ideal field condition.....	65
Figure 3-37. OSA digging control output in stony locations	66
Figure 3-38. OSA pH ISE field calibration	68
Figure 3-39. OSA nitrate ISE field calibration	68
Figure 4-1. Conceptual design of the apparatus.....	74
Figure 4-2. Manual soil chemical analyzer and sampler DAQ flow	76
Figure 4-3. Multiple ISEs manual probe operation.....	78
Figure 4-4. Tentacle noise test for the first Tentacle (100 ms sampling rate).....	80
Figure 4-5. Tentacle noise test for the second Tentacle (1000 ms sampling rate)	81
Figure 4-6. Effect of operator movement on the first Tentacle reading	81
Figure 4-7. The first Tentacle (100 ms sampling rate) cross talk test result	82
Figure 4-8. The second Tentacle (1000 ms sampling rate) channel 1 cross talk test result	83
Figure 4-9. The first experiment ISEs field calibration	85
Figure 4-10. The second experiment ISEs field calibration.....	86

Figure 5-1. DUALEM 21-S mapping location.....	92
Figure 5-2. DUALEM-21S filtering hierarchy	93
Figure 5-3. DUALEM-21S filtering GUI.....	94
Figure 5-4. Dualem 21-S response function.	98
Figure 5-5. The Brute-Force inversion GUI	99
Figure 5-6. DUALEM-21S Brute-Force EC _a inversion flow chart	101
Figure 5-7. Procedure of shallow muck soil layer volume and average depth calculation.....	102
Figure 5-8. FieldScout EC _a 110.....	103
Figure 5-9. Existing Spectrum EC sensor tip.....	103
Figure 5-10. Operation of the subsoil manual EC _a probe:.....	104
Figure 5-11. Subsoil Manual EC _a Probe calibration.....	104
Figure 5-12. Brute-Force EC _a inversion validation locations.....	105
Figure 5-13. Three-dimensional graph of modelled soil EC _a	106
Figure 5-14. Example of Brute-Force inversion output along transect	107
Figure 5-15. Depth of muck soil layer.....	108
Figure 5-16. Shallow layer EC _a inversion map	109
Figure 5-17. Deep EC _a inversion map	111
Figure 5-18. Soil EC _a profiles and models on locations with deep muck soil layer depth.....	112
Figure 5-19. Soil EC _a profiles and models on locations with shallow muck soil layer depth.....	114
Figure 5-20. Soil EC _a profiles and models on locations with penetration restricting depth	115
Figure 5-21. EC _a profiles and models on locations with moderate topsoil layer depth	116
Figure 5-22. Comparison of muck soil layer depth estimation	118
Figure 9-1. The OSA motors and actuators schematic wiring diagram	136
Figure 9-2. The OSA sensors schematic wiring diagram.....	137
Figure 9-3. The OSA DAQ schematic wiring diagram.....	138

Abbreviation

ACM	Agitated soil chamber module
CEC	Cation exchange capacity
CMOS	Complementary metal-oxide-semiconductor
DAQ	Data acquisition system
DC	Direct current
DI	Deionized (water)
DSM	Direct soil measurement
EC	Electrical conductivity
EC _a	Apparent soil electrical conductivity
EMI	Electromagnetic induction
GPS	Global positioning system
GUI	Graphical user interface
H ₂ PO ₄ ⁻	Dihydro phosphate ion
HCP	Horizontal coplanar
HDOP	Horizontal dilution of precision
H _p	Primary electromagnetic field
HPO ₄ ²⁻	Hydro phosphate ion
H _s	Secondary electromagnetic field
I/O	Input/output
ISE	Ion-selective electrode
LIN	Low induction number
NH ₄ ⁺	Ammonium ion
NMEA	National Marine Electronics Association
NO ₃ ⁻	Nitrate ion
OM	Organic matter
ORP	Oxidation-reduction potential
OSA	On-the-spot soil analyzer
PA	Precision agriculture
PRP	Perpendicular coplanar
PSS	Proximal soil sensing
PTFE	Polytetrafluoroethylene
PWM	Pulse width modulation
RMSE	Root mean squared error
RPM	Revolution per minute
RTK	Real-time kinematic
Rx	Receiver coil
Tx	Transmitter coil
UTM	Universal Transverse Mercator
VDC	Volt direct current
VRA	Variable rate application
WAAS	Wide Area Augmentation System

1. Introduction

1.1. General Introduction

Field investigation and soil sampling are crucial in the precision agriculture (PA) cycle as they provide the farmer with georeferenced information on what is happening in the field (Srinivasan, 2006). One of the challenges to the adoption of PA agronomically was limited sampling quantity and sensor robustness (Ferguson, 2019). The development of new sensors and mobile soil sensing platforms has mitigated the sampling and sensor issues. However, more efforts are needed to refine the existing field scouting processes and to provide tools for data analysis and interpretation in order to improve accessibility and to harmonize the adoption of PA methodology into agronomical practices (Jochinke et al., 2007; Bobryk et al., 2018).

Agricultural intensification through fertilization has become a common practice to increase crop yield (Matson et al., 1997; Fuglie, 2010). However, if fertilizer requirements are not adequately assessed through field investigation, fertilization may lead to various detrimental consequences, such as eutropication (Bouwman et al., 2017; Tirado, 2008) and an increase in greenhouse gas emissions (Van Noordwijk et al., 2017) due to uniform and over application. Therefore, understanding soil chemical properties in the field is essential to provide effective fertilizer application recommendations to farmers (Havlin et al., 2016; IPNI, 2012).

One of the methods PA offers to efficiently quantify soil chemical properties is direct soil measurement (DSM) (Adamchuk et al., 2005), which utilizes the ion-selective electrode (ISE) to analyze certain important soil chemical properties (e.g. pH, nitrate, phosphate and potassium) directly on the soil surface. ISE sensors require firm contact with the soil surface and considerable time is required to obtain a stable response. The DSM relies on natural soil water content in the field. This benefit eliminates the requirement to wet the soil sample to a specific soil-water ratio before measuring, thus, allowing for a simplified sensing platform design.

The need to conduct DSM at a specific location and to accommodate different farm scales require the development of two kinds of sensing platforms: automatic and manual. The automatic soil sensing platform is meant for large-scale farmers who want to use an implement-style platform attached to a small tractor or to a vehicle. The manual soil sampler is meant for a rough, general estimation of specific soil chemical properties and it is purposely designed for small-scale farmers. Both platforms should be able to accommodate several ISEs sensors for various soil chemical analysis at the same time, thus, providing rapid field investigation of multiple soil chemical properties.

Apparent soil electrical conductivity (EC_a) information can be used to support the operation of the soil sensing platforms by effectively selecting appropriate sampling locations (Heil and Schmidhalter, 2017). Soil EC_a is typically measured by an electromagnetic induction (EMI) sensor. Due to its mobility and on-the-go characteristics, soil EC_a measurement from an EMI sensor provides better spatial and temporal resolution as compared to traditional soil sampling (Corwin, 2008; Corwin et al., 2008).

The EMI reading represents bulk soil EC_a information. It needs to be inverted to get the layered soil EC_a response (Sudduth et al., 2013). As a common practice, soil EC_a data are inverted using commercially available software such as EM4Soil (EMTOMO, Lisbon, Portugal). Additionally, other standalone research scale inversion software have been developed, such as UBC-GIF (Geophysical Inversion Facility, Department of Earth and Ocean Sciences, University of British Columbia, Vancouver, Canada) and AarhusInv (Department of Geoscience, Aarhus University, Aarhus, Denmark). These software are often not freely accessible and require special skillsets to interpret the inversion results. Therefore, a simple soil EC_a inversion algorithm needed to be developed. The soil EC_a inversion result can be used to produce feedback for farmers on the appropriate farming operation (O'Connor, 2012) and to provide a reliable sampling scheme for further soil chemical investigation using soil sensing platforms.

1.2. Statement of Rationale

Traditional farmers and the agricultural industry often use high and uniform rate fertilization to boost their crop yield. The devastating effects of those practices are apparent. This problem is attributed to the inability to quantify available soil nutrients in the field. Soil measurements require consideration of the soil sensing platform design. Furthermore, determination of fertilizer demand generally relies on cumulative randomized grid sampling. It is necessary to develop the PA methodology to provide a better understanding of soil chemical attributes in the field. First, reliable automatic and manual soil sensing platforms are needed to accommodate multiple ISEs as well as different farmer scales. Both platforms need to be tested in different agricultural settings to thoroughly determine their efficacy. Secondly, a soil EC_a inversion is needed to provide information about important soil profile parameters to complement the soil sensing platforms operation.

1.3. Objectives of the Research

The ultimate goal of this research was to develop robust soil sensing platforms for measuring soil chemical properties in different agricultural settings.

The specific objectives of this research were:

1. To develop an automatic soil sensing platform for the measurement of various soil properties.
2. To develop a portable soil sensing system based on an ion-selective electrode for the measurement of soil chemical properties in the field.
3. To develop a soil EC_a inversion algorithm for determining key parameters of soil profile and appropriate sampling locations to support the operation of the soil sensing platforms.

2. General Review of the Literature

2.1. Soil Chemical Investigation

Agriculture has grown rapidly following the intensification of crop production through fertilization (Martinelli and Filoso, 2008; Clearwater et al., 2016; Matson et al., 1997). Unfortunately, heavy fertilization has detrimental consequences, one of which is eutrophication (Savci, 2012; Tirado, 2008). Moreover, excess phosphorus runoff and nitrogen fertilizer leaching causes groundwater contamination (Ding et al., 2014; Menció et al., 2016). The devastating effects, such as dying aquatic ecosystems and non-potable water, were obvious (Nagendran, 2011; Bunting et al., 2011). The agricultural sector has contributed to non-point emission sources which are difficult to quantify, track and manage (Hansen and Hansen, 2014). However, the negative effect of heavy fertilization can be mitigated through monitoring available soil nutrition to prevent excess fertilizing (Liu et al., 2015).

Variable rate fertilizer application (VRA) has become one solution to monitor existing soil nutrients as well as providing recommendations for the fertilization to maximize yield and prevent excess fertilizer application (IPNI, 2012; Havlin et al., 2016). VRA requires information on various soil properties and their corresponding location in the field. Soil properties, such as soil nitrogen (N), phosphorus (P), pH, organic matter (OM), electrical conductivity (EC), texture, and topographical information, are essential for VRA input parameters (Fleming et al., 2000; Baxter et al., 2003). These inputs can be processed, analyzed, and modelled to create fertilizer recommendation maps (Van Alphen and Stoorvogel, 2000; Dillingham et al., 2012). The quality of the maps can be improved using on-the-go soil measurements which provide high spatial and temporal resolution of soil properties data (Adamchuk et al., 1999; Adamchuk et al., 2004).

Various methods have been developed to determine important soil properties for VRA. From the PA perspective, a rapid, real time, robust and low cost sensing systems is essential to gather a substantial amount of georeferenced information (Srinivasan, 2006). The need to rapidly characterize soil properties for VRA is essential especially for mobile soil nutrient such as soil nitrate. As a standard practice, soil nitrate is determined through

wet chemistry involving a rigorous set extraction process and bulky, yet expensive, laboratory apparatus such as a steam distiller, ion chromatograph and spectrophotometer (Mulvaney, 1996).

Soil spectra analysis has provided a good alternative to overcome standard soil nitrate analysis limitations (Mulla, 2013). As an example of this, Jahn et al. (2006) used mid-infrared spectroscopy and wavelet analysis, Ehsani et al. (1999) utilized near infrared reflectance spectroscopy and partial least square regression (PLSR) and Yan et al. (2018) utilized a portable laser induced breakdown spectroscopy (LIBS). The coefficient of determination of field nitrate from those methods ranged from 0.59 to 0.98 with a standard error of 5.0 to 7.7 ppm nitrate. Although spectral analysis results were promising, the instruments are costly and need extra care for *in-situ* application. Moreover, distinct soil properties can have overlapping ranges of spectral signatures, which make it difficult to pinpoint the variability of specific soil properties which are of interest (Ge et al., 2011; Angelopoulou et al., 2020; Lee et al., 2009).

Another soil nitrate analysis, such as colorimetric, direct UV-absorbance, and chromatography, provided limited performance, lifetime and field application (Sah, 1994). The microchip capillary electrophoresis (MCE) sensor is relatively new and needs further development for direct soil nitrate measurement (Smolka et al., 2017). An ion selective field effect transistor (ISFET) technique has been developed for soil nitrate determination (Artigas et al., 2001). However, its limited lifetime, long term drift and ion interference limits ISFET application in analyzing soil chemical concentrations *in-situ* (Mapare et al., 2013).

The dynamics of soil nitrates depend on various factors such as bacteria activity, charge of soil minerals, cation exchange capacity (CEC), and bacterial nutrient availability (Truog, 1947; Peterson, 1982; Addiscott, 2005). All of these soil processes are affected by the soil pH level. Again, soil spectra measurement has become an appealing technique to rapidly characterize soil pH levels. Christy (2008) used on-the-go near infrared reflectance spectroscopy to predict soil pH in the field. The coefficient of determination was 0.62 to 0.68 with accuracy between 0.44 to 0.46 pH. Again, as outlined previously, a

mixed spectral signature of soil properties becomes an inherent problem for spectral analysis in the field (Mulla, 2013). ISFET sensors were also developed for *in-situ* soil pH measurement. However, the tedious field sampling preparation, electrode robustness and response instability limit the ISFET application for rapid soil pH determination (Viscarra Rossel and Walter, 2004; Lobsey et al., 2010).

Brouder et al. (2003), Davenport and Jabro (2001) and Sinfield et al. (2010) discussed a promising method for *in-situ* soil chemical analysis using ion-selective electrode (ISE) sensors. Moreover, laboratory soil pH measurement utilizes an ISE sensor as a standard method (Thomas, 1996). Depending on the type of its selective membrane, ISE can measure soil chemical concentrations in the field (Kuang et al., 2012). The advantages of ISE are that they are small and portable, with a relatively fast response and only need deionized water as a solvent. Hence, ISE was suitable for on-the-go soil measurements or grid sampling (Adamchuk et al., 2004; Kim et al., 2007; Domingue et al., 2005). In addition, ISE is also appropriate for direct soil measurement (DSM) using natural soil moisture (Conkling and Blanchar, 1989; Adamchuk et al., 1999). Michael et al. (2011) found that on-the-go soil pH measurement using ISE yields a decent coefficient of determination of 0.63 to 0.84 with an accuracy of 0.55 pH. Direct soil measurement of soil nitrate using ISE also yields a satisfactory coefficient of determination of 0.87 (Dhawale, 2015).

Figure 2-1 represents the ISE's potentiometric principle where the ISE and the analyte are in electrochemical equilibrium, meaning that the energy for detaching the electron from the electrode surface and the ion from the analyte solution are equal, therefore, preventing electrical current flow (Schoning et al., 1999; Skoog et al., 2014; Bakker, 2019). During measurement, the sensitive membrane on the tip of the electrode will either be reduced or oxidized. This process creates a potential difference between the sensitive membrane and the reference electrode. This potential can be measured using a high resistance voltmeter. Since the electrode and the analyte solution have a common ion, the potential difference depends on the analyte concentration and follows the Nernst equation (2-1).

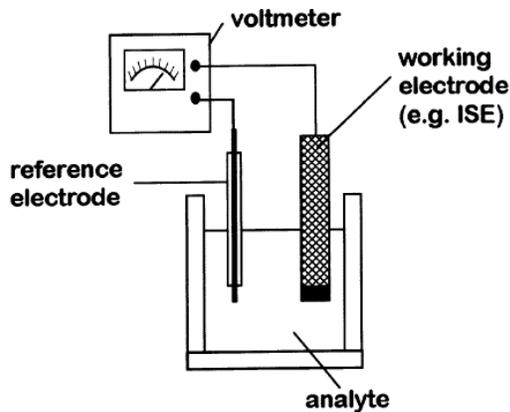


Figure 2-1. ISE's potentiometry circuit.
Adapted from Schoning et al. (1999)

$$\Delta E = \Delta E_0 + \frac{RT}{zF} \log a_s \quad (2-1)$$

where:

E = Measurable voltages (V)

E_0 = Standard Galvanic potential (V)

R = Gas constant ($8.31451 \text{ J K}^{-1} \text{ mol}^{-1}$)

T = Absolute temperature (K)

z = Charge number of ion

F = Faraday constant ($96,485.3329 \text{ C mol}^{-1}$)

a_s = Activity of the potential determining ion

Direct soil measurement using ISE provide various advantages compared to rigorous, time consuming and expensive wet chemistry analysis. The ISE's small and portable form factor offer minimal sampling footprint. The ability to estimate soil chemical concentration using natural soil moisture simplify the soil measurement process compared to soil extraction in the laboratory. Additionally, DSM using ISE provide a real-time measurement of soil chemical concentration *in-situ* which superior compared to standard laboratory analysis as it minimizes soil sample alteration due to biochemical processes.

2.2. Soil Sensing Platform

2.2.1. Automatic Soil Sensing Platform

An automatic on-the-go soil sensing platform was needed to rapidly acquire various soil data in a broad area. Compared to grid sampling, this method offers fewer overall errors since it can provide more data (Adamchuk et al., 2004; Adamchuk et al., 2007). One example of these systems is shown in Figure 2-2; it can take soil samples at 0 to 15 cm depths by cutting the soil using a saw blade. The soil sample was directed into the conveyor belt and metered using a scrapper blade to the intended height. The soil sample was placed in the nitrate analyzer section and dumped into an acrylic chamber and agitated. Then, the sample was sprayed with an adjustable quantity of water to get a soil mass to water ratio of 1:15 to 1:3. Then, the moist sample was analyzed using a nitrate ISE for 90 s. A confidence interval method to reduce the measurement time to 6 s with a 95% prediction accuracy was created (Adsett et al., 1999; Thottan et al., 1994). It was

clearly observed that inherent soil water content affected the nitrate ISE response time. The ISE voltage was acquired using a BCC52 microprocessor with an Intel 8255 chip which support 3 channel ISEs, each with 8-bit resolution. After the nitrate measurement, the sample was released through a valve underneath the chamber.

The water mixing system had difficulty in sensing inherent soil water content which would increase the overall measurement time. The saw blade soil sampler might not provide enough soil sample. Moreover, wet soil might makes it difficult to detach from the recess conveyor and could contaminate the next soil sample. Since the platform was fixed mounted to a tractor, it reduced the cross-platform operability. These drawbacks and other electrical problems were revealed in the field test.

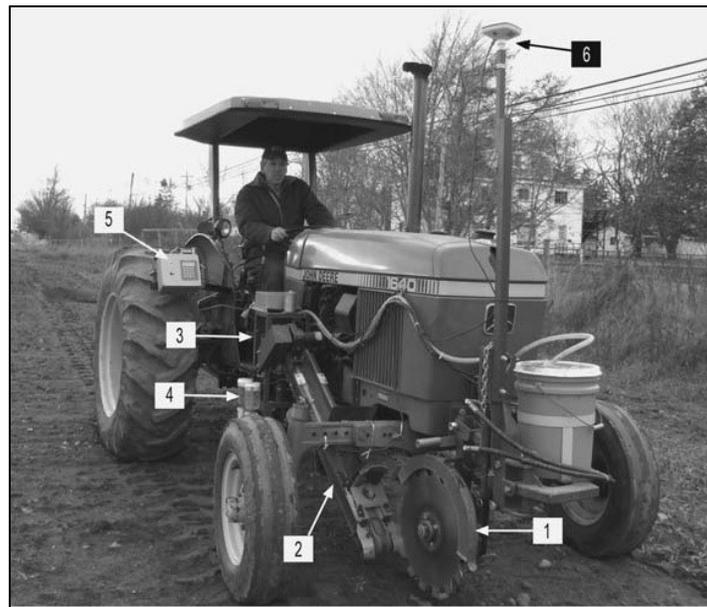


Figure 2-2. Tractor mounted on-the-go soil measurement.
(1) soil sampler, (2) soil metering and conveying, (3) nitrate extraction and measurement,
(4) auto-calibration, (5) control and (6) GPS. Adapted from Sibley et al. (2008)

Adamchuk et al. (1999) proposed a design where soil samples were collected using stainless steel shanks while moving (Figure 2-3). The sample was brought up from a 20 cm soil depth to contact the pH ISE. This measurement method utilized inherent soil water content (0.15 to 0.25 g/g) or as it is called, direct soil measurement. Moreover, it only took 5 s after the pH ISE insertion to obtain a stable measurement. The field test yielded a strong agreement between the ISE output and the standard laboratory test ($R^2 = 0.83$)

with a standard error of 0.45. Additionally, the sensing platform was intended as the tractor's three-point hitch attachment instead of fix mounted.

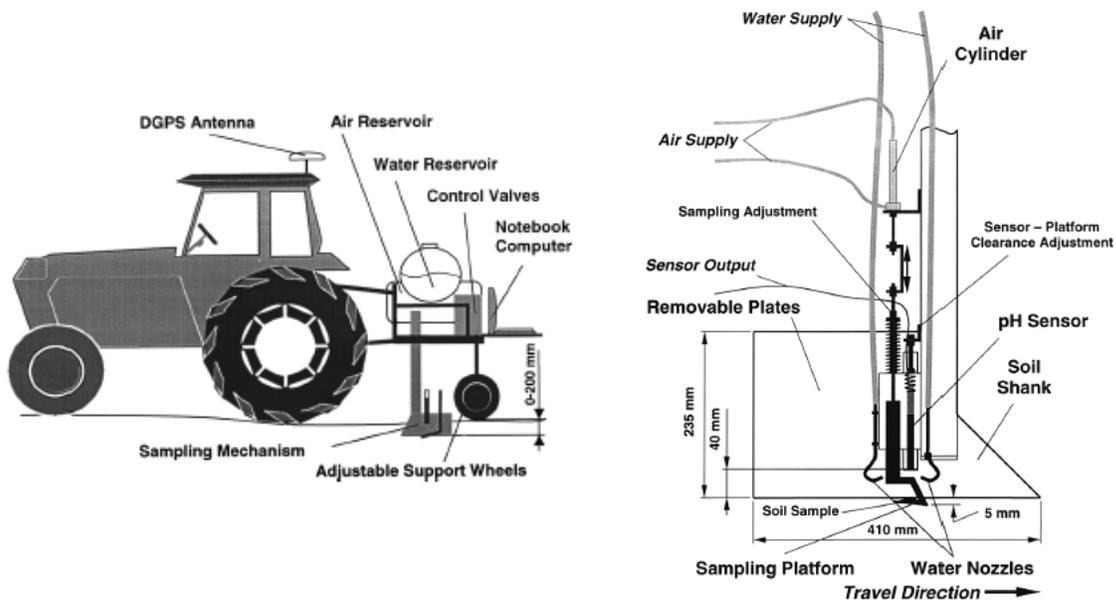


Figure 2-3. Schematic of the on-the-go platform.

Tractor mounted three-point hitch (left) and soil sampling mechanism for soil pH measurement (right).

Adapted from Adamchuk et al. (1999)

To accommodate various soil property measurements, the DSM technique was refined by adding multiple ISEs. Compared to laboratory measurements, the attempt resulted in good prediction for soil pH (R^2 of 0.93 to 0.96) but less for nitrate (R^2 of 0.41 to 0.51) (Adamchuk et al., 2005). The low prediction quality was mainly due to soil variability. This claim was in accordance with Mulvaney (1996), where delay in laboratory analysis might result in changes in nitrate and ammonium content due to activity of bacteria. Nevertheless, since the pH measurement yielded a favorable result, the platform was successfully commercialized with an additional contact soil EC sensor and one pH ISE for cross validation and error filtering (Adamchuk et al., 2007). The platform is known as Veris Mobile Sensor Platform (Veris Technologies, Inc. Salina, Kansas, USA).

In an effort to mitigate the poor prediction of soil nitrate, potassium and sodium, Sethuramasamyraja et al. (2008) introduced an agitated soil chamber module (ACM) on a towed platform (Figure 2-4). After sampling, the soil was brought up into the ACM. Then, the soil sample was agitated and sprayed until 1:1 (volumetric) soil paste was achieved.

The chamber, blade and agitator shaft were made from plastic to avoid electrical noise. ACM follows the flow injection analysis by letting the agitated soil solution flow into the analysis chamber which was similar to that of Cardwell et al. (1988) and Dimitrakopoulos and Dimitrakopoulos (2001). Unfortunately, the ACM method did not provide better results than the previous DSM design with R^2 of 0.85 to 0.89 for soil pH, 0.50 to 0.54 for potassium, and 0.14 to 0.32 for nitrate. The researchers pointed out that the use of more affordable half-cell ISEs and the design of ACM might cause the low prediction quality.

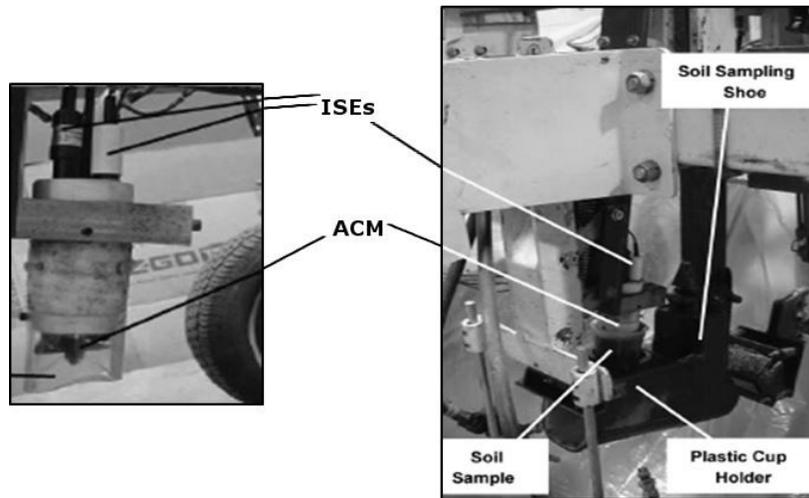


Figure 2-4. Agitated soil chamber module (ACM) arrangements.
Adapted from Sethuramasamyraja et al. (2008)

An interesting development of an on-the-go soil sensing platform determining soil liming requirement came from Viscarra Rossel et al. (2005). They built a tractor implement type with a three point hitch soil sensing platform. This consisted of a tine to loosen the soil, an extraction fan and a cyclone combination to aspire and pulverize the soil, a 2 mm sieved, a soil metering unit (4 cm^3), a chamber for mixing the soil sample with analytical solution (e.g. 0.01 M CaCl_2 , water or Mehlich lime buffer), agitator, pH ISFET and washing nozzles (Figure 2-5). The total measurement cycle was 27 s which consisted of soil sampling and sieving, soil volumetric measurement, mixing and pH measurement, and washing which each required 3, 10, 10 and 4 s, respectively. Spatial resolution was 75 m between sampling points in a transect. Large variations between the measured and standard laboratory pH tests were found during the field trial. The ISFET response was slow and had a low accuracy (0.37 to 0.68 pH).

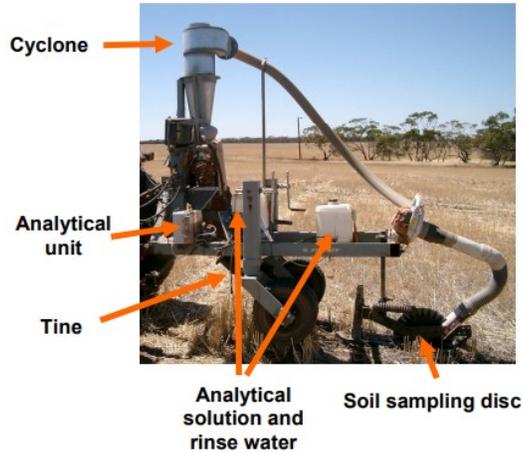


Figure 2-5. On-the-go soil pH and lime requirement measuring system.
Adapted from Viscarra Rossel et al. (2005)

Different soil sensing platform alternatives using near infrared reflectance spectroscopy were presented by Christy (2008). The main components of the near infrared spectrophotometer soil sensing platform were a disc coultter for cutting the plant residue and a shank with incorporated spectrophotometer (Figure 2-6). The platform was attached to tractor's three point hitch and pulled at 6 km/h. The depth of measurement was 7 cm with 20 Hz soil spectra sampling rate. Soil pH prediction was relatively low ($R^2 = 0.46$) with an accuracy of 0.53 pH. Better results were determined from organic matter prediction ($R^2 = 0.67$) with an accuracy of 0.52% organic matter content. As described earlier, a problem with the overlapped spectral signature made it difficult to delineate the specific soil properties variation.

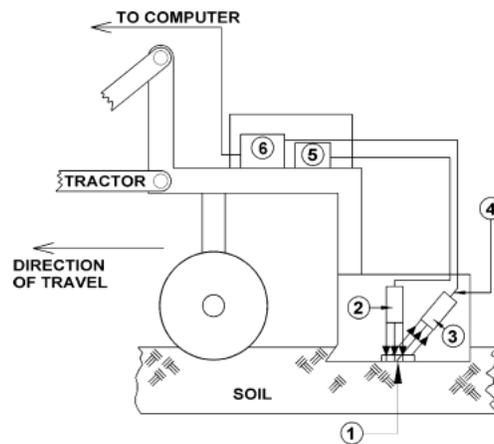


Figure 2-6. On-the-go soil sensing platform based on near infrared spectrophotometer.
(1) Sapphire window; (2) halogen lamp; (3) collection optic; (4) fiber optic;
(5) spectrometer; (6) power supply. Adapted from Christy (2008)

Another soil spectra based soil sensing platform was presented by Kodaira and Shibusawa (2013). This time, visible-near infrared sensor was incorporated into a shank and pulled behind a tractor using a three point hitch (Figure 2-7). The measurement depth was from 5 to 35 cm. They found good prediction for soil pH ($R^2 = 0.69$) but lower for soil nitrate ($R^2 = 0.45$). From the field trial, several issues on utilizing soil spectra were found. They included the fact that the sensor picked up unwanted materials such as plant residue, stone and larva, which contribute to the difficulty in quantifying the specific soil spectra signature. Often, the sensor's housing was filled with soil especially when crossing the field furrows.

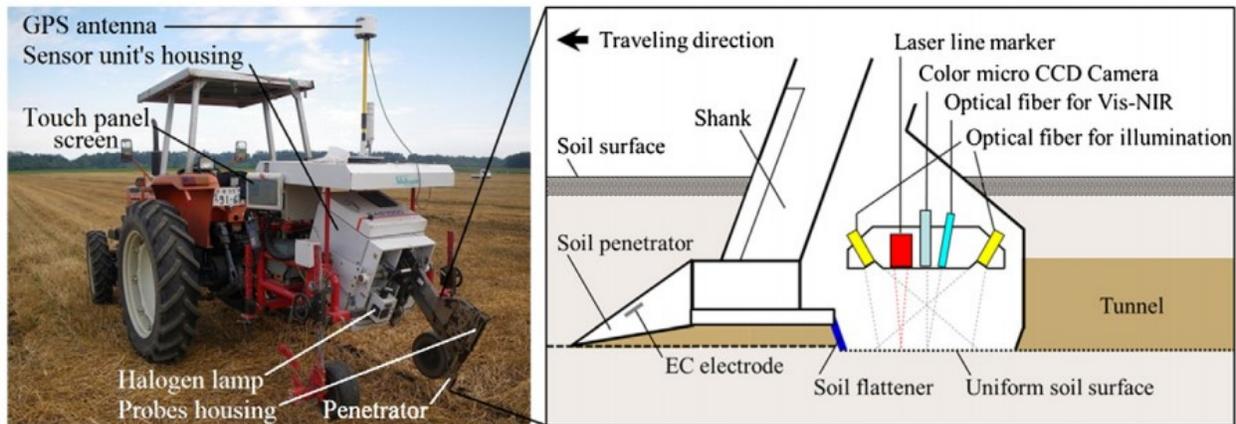


Figure 2-7. On-the-go soil sensing platform based on visible-near infrared sensors.
Adapted from Kodaira and Shibusawa (2013)

A fully autonomous soil sensing platform was developed by Yan et al. (2020). The platform was inspired by the ExoMars rover. The four independently driven wheeled platform has a drill-core soil sampling mechanism, on-board nitrogen analyzer using LIBS and an autonomous navigation sensing system (Figure 2-8). The maximum sampling depth was set to 50 cm to compensate for the small platform payload. Future field tests are needed to assess the platform's performance in typical agricultural settings.

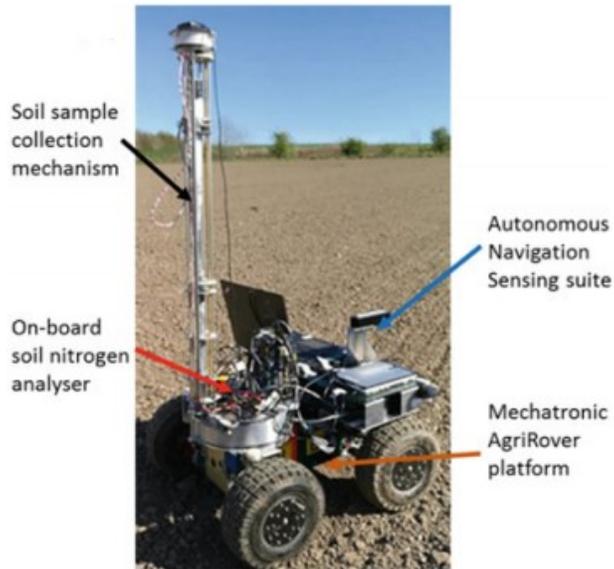


Figure 2-8. AgriRover: an autonomous soil nitrate sensing platform.
Adapted from (Yan et al., 2020)

With advances in geostatistics, sparse or systematic grid soil sampling data can be interpolated to create a map of soil properties of interest (Oliver, 2010). On-the-go soil sensing platforms are unable to efficiently conduct sparse or grid sampling because the platform operates in a transect arrangement. Adamchuk et al. (2014) developed an on-the-spot soil sensing platform called an OSA (on-the-spot soil analyzer) to alleviate that problem. The platform was required to perform a quick stop at certain sampling locations to make measurements. First, the soil was dug using a rotary blade. Then, nitrate and pH ISEs were deployed firmly onto the soil and this was followed by the measurement. Lastly, while moving forward, the soil from the digging action was returned to the measurement hole by a steel guard. The total time for the whole process was around 60 s. Since the total cycle time was relatively short, the on-the-spot method could deliver the same sampling density as the on-the-go method. The system was successful in determining soil nitrate and pH with a coefficient of determination of 0.87 and 0.84, respectively. The platform used expensive NI-MyDAQ and NI-USB-6216 data acquisition cards (National Instruments, Austin, Texas, USA). Moreover, the platform did not have a digging depth control. So, it was only able to perform single depth measurements. Importantly, the digging action still relied on a series of switches, which are inefficient in terms of energy usage and not adaptive for different terrain conditions.

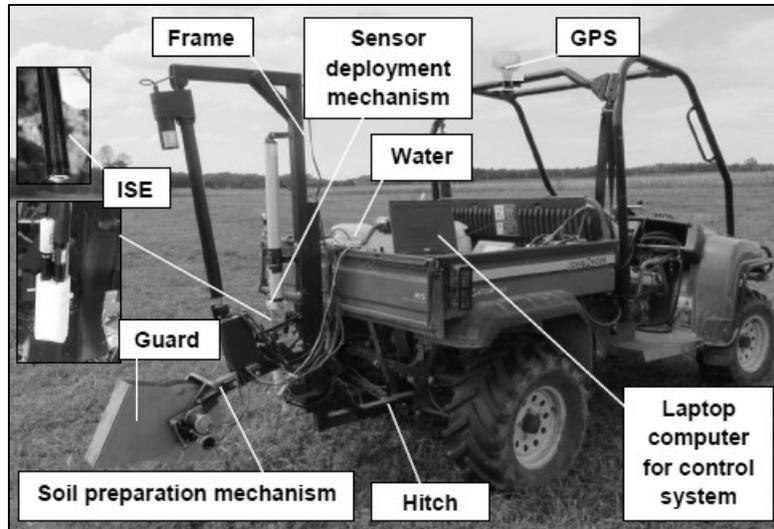


Figure 2-9. On-the-spot soil analyzer.
Adapted from Dhawale (2015)

2.2.2. Manual Soil Sensing Platform

Depending on the sensors used, on-the-go soil sensing platforms were developed to quantify specific soil properties, such as soil compaction, water content, depth of plow pan, electrical conductivity and organic matter content (Viscarra Rossel et al., 2011). They were usually pulled by a vehicle (tractor or all-terrain vehicle) and operate in a transect setting (Adamchuk and Viscarra Rossel, 2010). A portable manual soil sensing platform equipped with a soil sampler mechanism for laboratory analysis is essential for low quantity soil sampling and for acquiring a general overview of specific soil properties in the field (Shibusawa, 2006).

Generally, the manual soil sensing platform follows a portable benchtop style instrument. Birrell and Hummel (2001) developed a multi ISFETs flow injection analysis (FIA) instrument for measuring soil nitrate (Figure 2-10a). From a manually extracted soil solution, the instrument successfully characterized the soil nitrate concentration with a coefficient of determination of 0.90. The sensor's fast response time (1.25 s) was advantageous for analyzing a large number of samples. They also developed an automated soil extraction system for field application (Figure 2-10b). The soil sample was inserted through the soil inlet. Then, the pneumatic plunger and slider push the soil into the rotating chamber. An electric stepper motor rotates the chamber to the extraction

station to get sprayed with the extracting solution. After another 45° rotation, the soil solution was discharged into the filtering cylinder. The filtered soil extractant was pumped to the FIA injection valve for analysis. When tested using the automated soil extraction system, the multi ISFETs FIA performed unreliably because of incomplete mixing between the soil sample and the extraction solution. The soil solution filter often clogged, resulting in an inconsistent flow of extractant to the ISFETs. A different soil extraction system was further developed to improve the incomplete mixing and inconsistent extractant flow. This time, the extraction solution was pumped through a soil core and the extractant was transferred using a suction pump to the injection valve for analysis using ISFETs (Figure 2-11). A comparison with the standard laboratory colorimetry method found that the new soil extraction system produces a better prediction of soil nitrate with a coefficient of determination of 0.72.

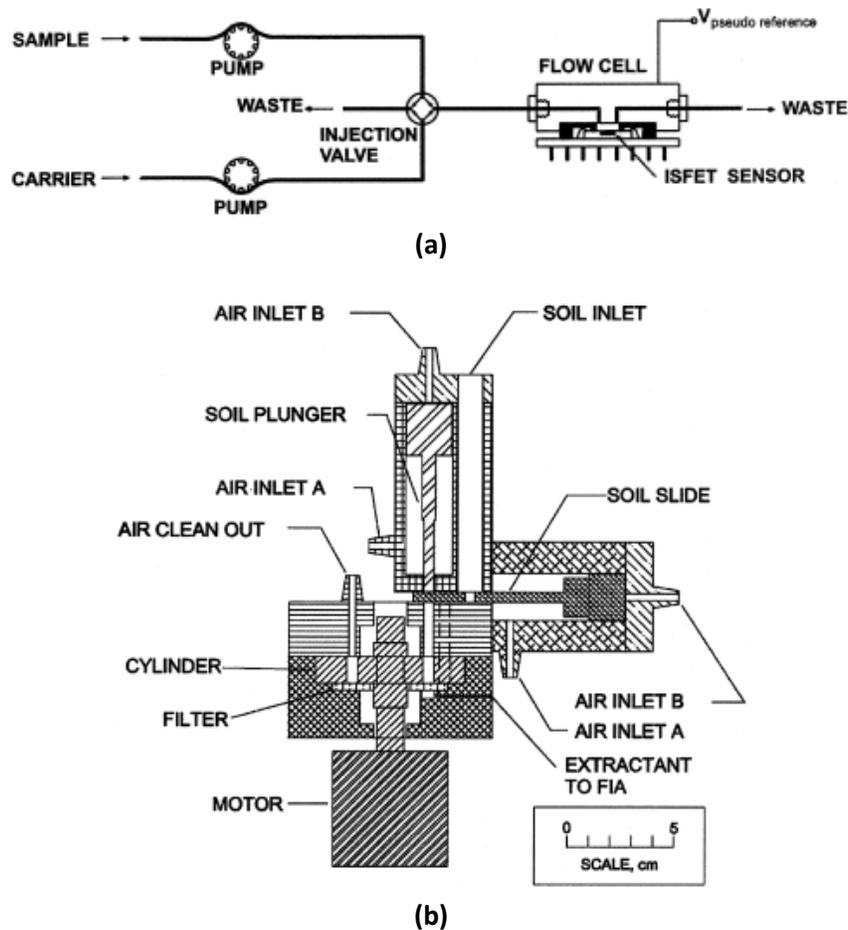


Figure 2-10. Multi ISFETs flow injection analysis apparatus. (a) schematic and (b) automatic soil solution extraction system. Adapted from Birrell and Hummel (2001)

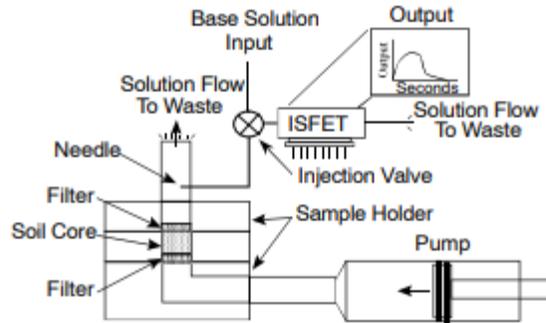


Figure 2-11. Soil core nitrate extraction system and analyzer using ISFETs.
Adapted from Price et al. (2003)

An ISE based benchtop soil analyzer was developed by Miao et al. (2008). This prototype consists of a series of solution containers to accommodate calibration solutions or soil extractant. The test chamber consists of two separate clear plastic materials which were specifically built to hold three mono indicator ISEs and a reference. Before measuring, the ISEs need to be calibrated by subsequently pumping a series of appropriate calibration solutions into the test chamber and then, measuring the ISE's response. The soil extractant measurement started by pumping the soil extractant into the test chamber. Then, the automatic fluidic control routine will log the electric potential from the ISEs and dispense the soil solution after measurement. The drawback from this system was the soil sample needs to be extracted first in order to operate. This limits the applicability of the system for *in-situ* measurements.

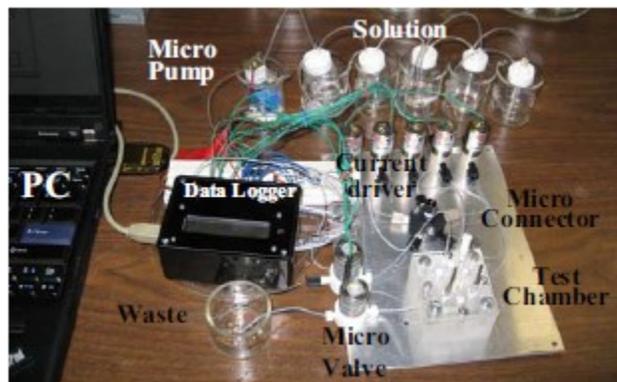


Figure 2-12. ISEs based benchtop soil analyzer.
Adapted from Miao et al. (2008)

Li et al. (2019) developed another ISE based benchtop soil nitrate analyzer complete with a soil solution extraction system for *in-situ* applications (Figure 2-13). In this system,

the soil sample needs to be collected manually. Then, the soil sample was weighed using onboard scales, mixed with deionized water and stirred. Next, the soil slurry was placed in a tube for centrifugation (40 s). Then, the ISE's measurement can be started after the supernatant was clear to minimize the interference from the soil slurry. Overall, the measurement cycle took 4 to 5 minutes per sample. A commercial benchtop soil nitrate analyzer called 360 SOILSCAN (360 Yield Center, Morton, Illinois, USA) is available in the market. Similarly, the system requires manual soil sampling. Then, two scoops of sampled soil need to be placed in measuring cup. The system will mix the sampled soil with water (1:1 volumetric), stir and conduct the nitrate and pH measurement using ISEs. The system has a built in software for tablet computer to ease the user interface, step by step instruction for ISE calibration and measurement, ISE data management and interpretation.



Figure 2-13. Benchtop *in-situ* soil nitrate analyzer using ISEs with soil extraction system. Adapted from Li et al. (2019)

Commercially available instruments with different sensing approaches were also used for soil measurement. Harmon et al. (2005) utilized a benchtop style LIBS (ADA Technologies, Inc., Littleton, Colorado, USA) for detecting lead contamination *in-situ* (Figure 2-14). The system required a rigorous sampling procedure which included soil digging, sieving and pressing the soil sample in an aluminum dish using a small hydraulic press. LIBS measurement was done by firing the laser to several locations on the sample

surface. Rogovska et al. (2019) used a 4100 Exoscan Fourier Transform Infrared (FTIR) spectrometer (Agilent Technologies, Inc., Santa Clara, California, USA) equipped with Diamond-Attenuated Total internal Reflectance (D-ATR) to determine the soil nitrate concentration. The system could successfully estimate soil nitrate concentration with a coefficient of determination of 0.83 to 0.90 and an accuracy of 8.3 to 8.8 mg nitrate/kg soil. Davenport and Jabro (2001) used an ISE based handheld Cardy Meter (Spectrum Technologies, Inc., Aurora, Illinois, USA) to estimate soil nitrate, potassium, sodium and pH using direct soil measurement. They found that the instrument was only reliable for pH estimation on moist soil ($\geq 15\%$ gravimetric water content).

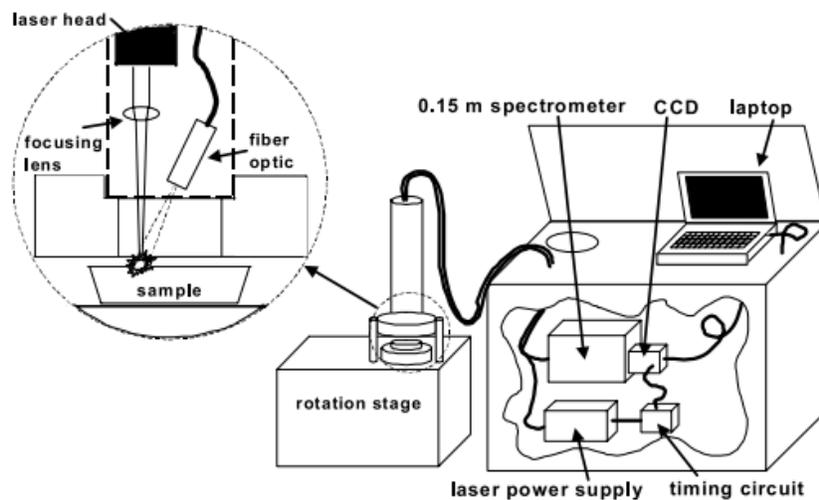


Figure 2-14. LIBS based *in-situ* soil analyzer.
Adapted from Harmon et al. (2005)

The aforementioned soil sensing platforms are generally bulky, expensive, and require tedious soil sampling and solution extraction procedures. To simplify *in-situ* soil sensing operation, a manual platform based on a modified generic soil sampler (JMC, Clements Associates Inc., Newton, Indiana, USA) was developed (Adamchuk, 2005). Portability and simplicity have become a major design factor to fulfill the need of farmers with small scattered fields and to accomplish the goal as an initial soil measurement tool. The Data Acquisition System (DAQ) was based on a commercial FieldScout pH 110 Meter (Spectrum Technologies, Inc., Aurora, Illinois, USA). This DAQ type was expensive and only supports one ISE sensor (Figure 2-15). Similar to standard soil core sampling, the operator pushes the platform into the soil. Then, the ISE was brought into contact with

the sampled soil to conduct direct soil measurement of the soil property of interest. The operator can monitor the reading stability from the DAQ display. Finally, manual pumping action sprays water towards the ISE to provide a cleaning action.



Figure 2-15. Manual probe for on-the-spot soil measurement.
Adapted from Dhawale (2015)

An affordable and multichannel DAQ needs to be developed to provide economic benefit for farmers as well as supporting multiple ISEs measurements. Various efforts have been conducted to fulfil those requirements. For example, Wang et al. (2011) used an 8 channel ISE DAQ system consisting of a CMOS analog multiplexer switch DG407, high impedance-low input bias amplifier INA116, 16 bit ADC AD7705, and ultra-low power MSP430F1611 MCU. A voltage follower circuit may also be added to further improve the amplifier impedance (Jianhan et al., 2007). Nowadays, the popular and affordable Arduino Uno microcontroller based on the ATmega328P chip, provided an opportunity to develop a simpler and more reliable multichannel DAQ.

2.3. Soil EC_a Inversion

Effective soil sampling schemes can be arranged to optimize soil sensing platforms operation in the field by selecting appropriate sampling locations from apparent soil electrical conductivity (EC_a) map (Corwin, 2008; Heil and Schmidhalter, 2017). In order to successfully produce soil EC_a maps, soil EC_a data need to be inverted. The results from the EC_a inversion are bulk soil slices with depth and their corresponding soil EC_a value. Knowing this information is crucial, not only for soil sensing platforms operation, but also to characterize the spatial distribution of soil properties with depth.

Proximal soil sensors, such as electromagnetic induction (EMI), can provide respectable spatial coverage and temporal information about the soil (Corwin and Lesch, 2003). The sensors measure bulk soil EC_a which is viewed as an important indicator of various soil properties (Bronson et al., 2005; Wienhold and Doran, 2008). Furthermore, due to their portable design and ease of operation, they have become a popular way to rapidly characterize soil (Doolittle and Brevik, 2014; Triantafilis et al., 2011; Saey et al., 2008; Allred et al., 2008).

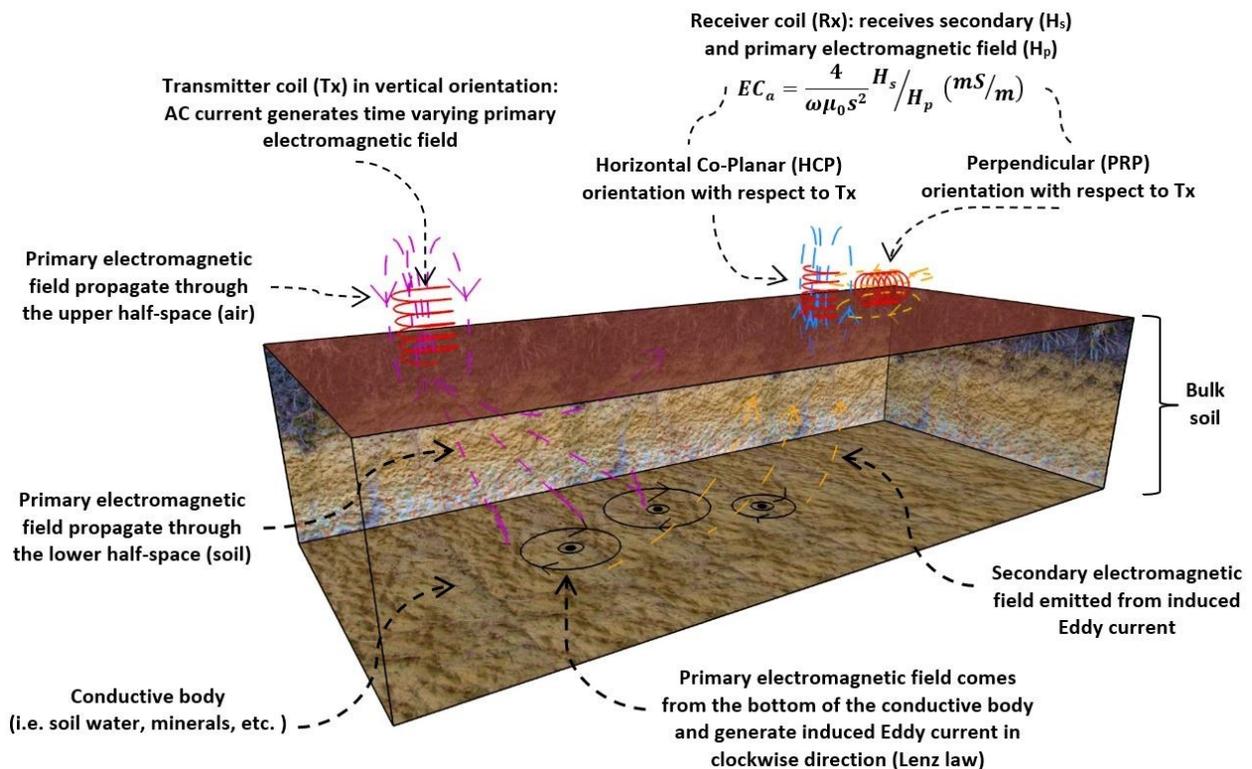


Figure 2-16. Typical electromagnetic induction (EMI) sensor working principle

Referring to Figure 2-16, a transmitter coil (T_x) will generate a primary electromagnetic field (H_p) resulting in an induced eddy current at the conductive body in the soil. This current will create a secondary electromagnetic field (H_s). Then, the receiver coil (R_x) will obtain the incoming primary and secondary electromagnetic fields. Apparent soil electrical conductivity (EC_a) is a function of T_x/R_x orientation and conductive materials which alter the phase, amplitude and direction of the transmitted field (Daniels et al., 2008).

The interpretation of the EC_a reading is not straightforward and it is often site specific (Bronson et al., 2005; Pedrera-Parrilla et al., 2016). Conduction in soil can be affected by various factors, such as soil water content (Brevik et al., 2006), clay content (Sun et al., 2011), soil temperature (Padhi and Misra, 2011), mineralogy (McNeill, 1980a) and salinity (Corwin and Lesch, 2003). Therefore, cross validation with standard laboratory measurements together with expert skills are essential to provide reliable inference (Doolittle and Brevik, 2014).

The EC_a measurement from EMI sensors represents a bulk cumulative soil EC_a response from each of the specific soil properties at specific depths. The contribution of these particular layers to the total EC_a reading often becomes an object of interest for both farmers and researchers (Saey et al., 2012a; Sudduth et al., 2013; Korsaeath, 2008). Inversion is a process to obtain the contribution of each depth layer to the bulk soil EC_a . According to Zhdanov (2015), inversion is an ill-posed problem meaning that it may have no solution because of too many parameters or too much noise. Ill-posed problems are commonly solved by regularization which considers some well-posed problems, followed by stabilization through the selection of the best possible solution. In general, there are two approaches to invert the EMI measurements into depth specific soil EC_a information: the finite element (Figure 2-17) and fixed slice cumulative depth response approach (Figure 2-18).

The finite element method is started by discretizing soil into 2-dimension rectangular meshes with two boundary conditions: a homogeneous external region given by the initial soil EC (electrical conductivity) model and fixed EC_a calculation given by the partial derivatives of the EMI's cumulative EC_a response function (Sasaki, 1989). By defining these two boundary conditions, the regularization process is fulfilled and thus, allows for forward EC_a calculation. The stabilization and model smoothing process is satisfied by optimizing the misfit function through the Levenberg-Marquardt method (Constable et al., 1987). The damping coefficient (λ) in this method controls the model's roughness which provides more realistic soil EC_a trends across soil depth and it is useful for delineating very stratified and high EC_a soils.

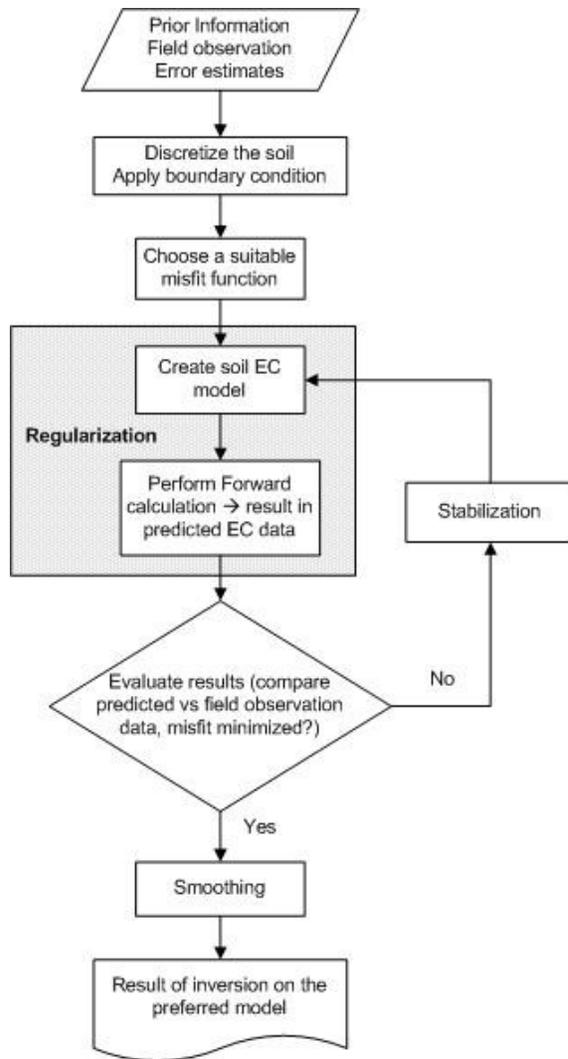


Figure 2-17. Finite element EMI inversion flowchart

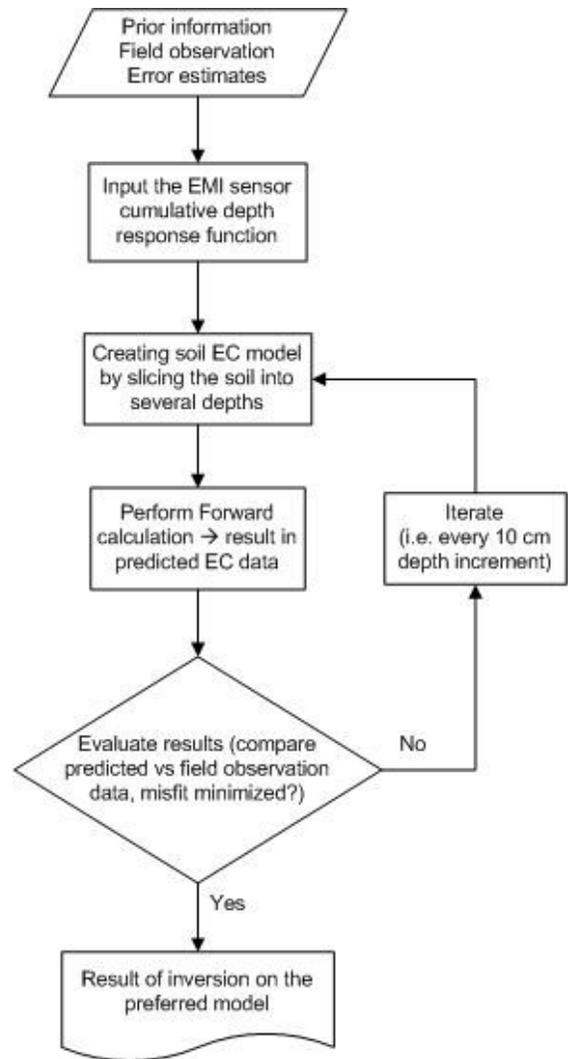


Figure 2-18. Fixed slice cumulative depth response EMI inversion flowchart

A major improvement on the algorithm was achieved by the development of a “full solution” to solve non-linearity between the model’s response and measured parameters beyond the EMI sensor’s low induction number (LIN) of 100 mS m⁻¹ range (Triantafilis et al., 2012; Santos et al., 2010b). However, the necessity of using a full solution was not essential for the general farming setup as it introduces more complexity in the algorithm and also because typical soil EC is between 0.8 and 100 mS m⁻¹ (Callegary et al., 2007). Nevertheless, the algorithm has the potential for commercialization as an off-the-shelf inversion software (EMTOMO, 2012) and has gained popularity among the scientific community (Triantafilis et al., 2011; Triantafilis et al., 2013; Davies et al., 2015).

The finite element inversion approach has been used extensively with various EMI sensors, such as EM34 (Santos, 2004), and Dualem 421 (Huang et al., 2016; Triantafyllis et al., 2011; Santos et al., 2010a). The application of the inversion algorithm was mainly used to delineate root zone, subsoil and vadose zone, and to characterize high salinity soil profiles (Davies et al., 2015; Zare et al., 2015). Generally, the method generates a better correlation between the inverted EC_a results and the measured soil properties if used in combination of multiple EC_a data from various EMI sensors with a different depth of investigation (Triantafyllis et al., 2013; Triantafyllis and Monteiro Santos, 2013; Triantafyllis and Santos, 2010, 2009).

Despite its success, the algorithm has some drawbacks primarily from the initial soil EC model determination where users need to define the bulk soil model with appropriate EC_a values and depths. This routine is cumbersome and becomes laborious even for an advanced user. Therefore, additional field experimentation is needed to provide a better starting model to accelerate the inversion process. This will be achieved by using some well-spaced ground truth measurements through a bore hole or laboratory soil analysis or alternatively, employing artificial neural networks or deep learning (Zhu et al., 2012).

The fixed slice cumulative depth response approach started with the determination of several fixed soil depth slices. Depending on the sensor used, the maximum slices are usually one less than the maximum EMI sensor configuration. Then, EC_a forward calculations were conducted using the EMI cumulative response and modelled EC value at a specified depth slice. Subsequently, the forward calculated EC_a results were compared with the measured EC_a from the EMI sensors to assess the misfit value. The initial soil EC model was further iterated with a fixed depth step (e.g. every 1 cm) until it reached a specified iteration number (Sudduth et al., 2013). Finally, the best model was selected with the smallest misfit value. This inversion method has been used extensively for archaeological mapping (Saey et al., 2008; De Smedt et al., 2013; Saey et al., 2012b), and to detect the depth of clay and ploughing layers (Saey et al., 2012a; Saey et al., 2009).

The fixed slice inversion is relatively simple and easy to use but has several shortcomings. The inversion result is a 1-dimensional data point consisting of information about soil slice depth and its EC_a value. Therefore, to create a transect depth or surface map, the inverted data points need to be interpolated using a geostatistical approach (e.g. kriging). Furthermore, the fixed slice inversion has a limited quantity of soil EC model slices. Thus, producing a distinctive soil EC layering which might not be as apparent as in the real condition. The finite element inversion utilizes Occam's algorithm and mesh displacements to create a 2-dimensional interpolation which produces a more realistic representation of soil EC than the former (Figure 2-19).

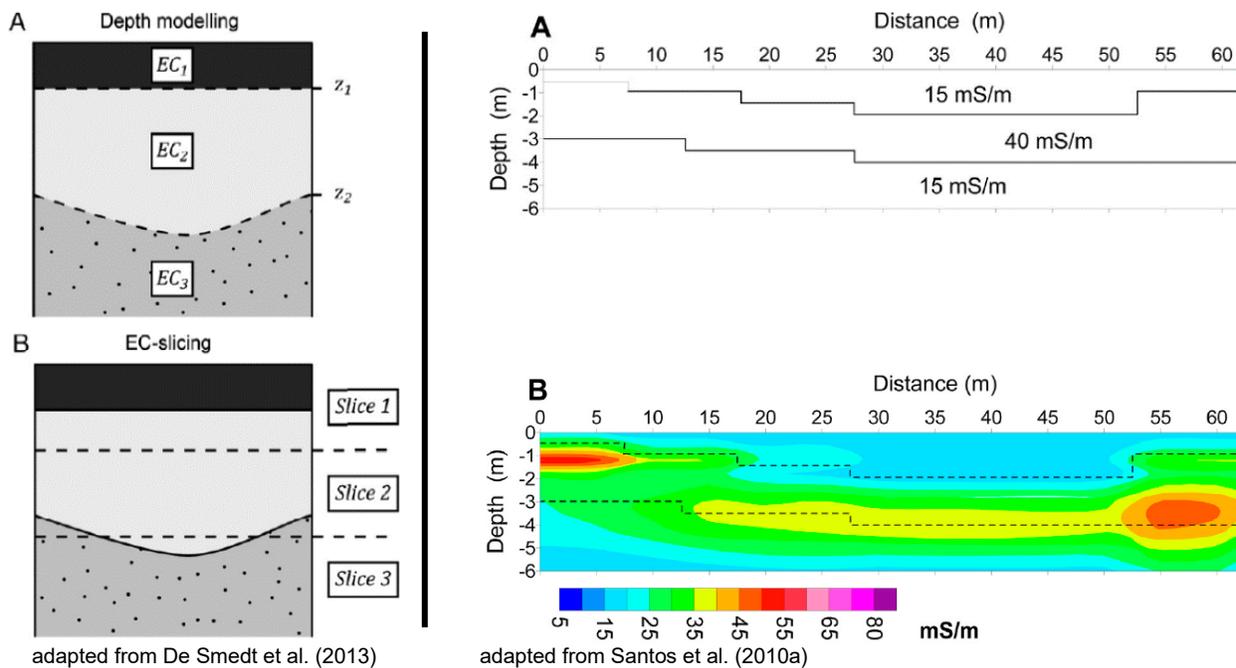


Figure 2-19. Difference in soil EC model (A) and inversion result (B) between fixed slice cumulative depth response (left) and finite element (right) approach

The soil EC_a inversion will provide information about contribution of specific soil properties at specific depths. Therefore, it will assist the farmers to understand the physical soil attributes of their field as well as its relationship with soil chemical properties especially in prolonged uniform field management. The soil EC_a inversion results can be used to determine soil properties associated with soil buffering capacity (CEC, texture, organic matter content) (Doolittle and Brevik, 2014; Adamchuk et al., 2007). Thus, together with the soil acidity information collected from DSM of soil pH using soil sensing

platforms, important agricultural management decision such as soil liming can be determined. Furthermore, from soil EC_a inversion maps, appropriate sampling locations can be selected for optimum soil sensing platforms operation.

Connection to Chapter 3

The on-the-spot soil analyzer (OSA) platform developed by Dhawale (2015) used an expensive commercial data acquisition system (DAQ). Moreover, the platform did not have a digging depth control. Thus, it was only able to perform a single depth measurement. Importantly, the digging action relied on a series of switches, which were inefficient in terms of energy usage and they could not be adapted for different terrain conditions. Additionally, the digging movement was in a curvature, which made the depth adjustment difficult. A new OSA platform design was developed in Chapter 3. A control system was incorporated into the new OSA design to efficiently control the soil sampling routine. Also, the Arduino Uno based DAQ system was developed to perform multiple soil chemical measurements using ISEs. The research was published in Proximal Soil Sensing conference 2019 in Missouri, USA.

Leksono, E., V. Adamchuk, M. Leclerc, R. Buelvas, C. Miller, and J. Park. 2019. Development of an on-the-spot proximal soil sensing platform for subsurface measurement of soil properties. In: Proceedings of the 5th Global Workshop on Proximal Soil Sensing, 27-31 May 2019, Columbia, Missouri, USA, eds. K.A. Sudduth, N.R. Kitchen, and K.S. Veum, 205-210. Columbia, Missouri, USA: USDA-ARS.

3. Development of an Automatic Soil Analyzer

Abstract

Proximal soil sensing (PSS) has become the preferred method to quantify various soil properties in situ. PSS utilizes portable soil sensors which can be mounted on a mobile platform, commonly called an on-the-go soil sensing operation. With the development of geostatistics, representative soil sampling locations (spots) can be selected effectively, thus, reducing the need to sample the entire field. The objective of this research was to develop a robust mobile soil sensing platform for on-the-spot soil measurements. The developed on-the-spot soil analyzer (OSA) platform was able to clean the soil surface for better sensor contact; it then covered the sampling hole automatically. The OSA used direct soil measurement (DSM) principles to quantify soil chemical properties using ion-selective electrodes (ISEs). The main physical design feature was the adoption of two sets of parallel linkages which were used to reduce the digging curvature. Additionally, a linearly actuated steel fender was used to cover the sampling hole. Various control design approaches and field tuning have been explored to develop robust digging control parameters. The OSA successfully conducted DSM of multiple soil chemical properties. The digging control was able to control the digging load accordingly and as a result, the average digging time was less than 25 s at 15 mm depth. The complete operation time was around 60 s for each sampling operation. Both pH and nitrate ISEs performed satisfactorily with R^2 of 0.59 and 0.72, respectively with RMSE of 0.43 pH and 0.16 pNO₃.

Keywords: On-the-spot soil analyzer (OSA), mobile sensing platform, direct soil measurement (DSM), PID controller

3.1. Introduction

Proximal Soil Sensing (PSS) frequently involves the use of portable soil sensors mounted on a mobile platform which is called on-the-go soil mapping. This operation requires a reliable platform to hold various sensors and it must be dependable for use in various field conditions. Various on-the-go platforms have been developed in recent years. For example: a vehicle mounted movable shank for direct pH measurement (Adamchuk et al., 1999; Adamchuk et al., 2007) and a tractor mounted conveyor-like soil sampler with saw blade to cut the soil, a water metering unit, a nitrate ISE, a controller and a GPS for soil nitrate measurement (Sibley et al., 2008). The aforementioned platforms produced a strong correlation between the pH ISE output and a standard laboratory test. Following the development of single pH ISE platform, a multiple sensor comprised of nitrate (NO_3^-), potassium (K^+) and sodium (Na^+) ISEs was developed with reasonable results (Adamchuk et al., 2005; Sethuramasamyraja et al., 2008).

Adamchuk et al. (2014) successfully patented an on-the-spot soil sensing system called on-the-spot soil analyzer (OSA). It was comprised of relatively small sized equipment and it could perform a fully autonomous operation. The platform needs to make frequent stops at the designated sampling locations to perform direct soil measurements (DSM). In the field test, the OSA was successful in determining soil pH and nitrate using ISEs with R^2 of 0.84 and 0.87, respectively. In comparison with the on-the-go approach, the on-the-spot soil analysis has the advantage of true *in-situ* measurements using natural soil moisture and requires a relatively short operation time.

Unfortunately, the OSA did not have a digging depth and power management control. Consequently, it is only able to perform single depth measurements and consumes a considerable amount of energy when digging. Moreover, the digging action relies on a series of switches, which are inefficient and not adaptive for use on different terrain conditions. Therefore, in this research, a new OSA platform was to be developed that would efficiently prepare the soil, perform multiple ISE measurements and restore the effect of the soil disturbance. Additionally, a control system was needed to improve the platform's energy efficiency and overall sampling time.

3.2. Materials and Methods

3.2.1. OSA Operation

The new OSA system needed to automatically measure the chemical properties on the soil surface. This operation consists of several actions. First, there is the soil preparation stage where it should be able to clean up the unwanted materials (e.g. plant residues) and smooth the soil surface at operator defined depths. Subsequently, the system should be able to maintain its stability while automatically deploying the ISEs to measure soil chemical properties. Lastly, it should be able to rinse the ISEs and cover the sample hole left from the soil preparation stage. Importantly, the operator should be able to control all of those actions conveniently from the operator compartment. The OSA operation flow can be seen in Figure 3-1.

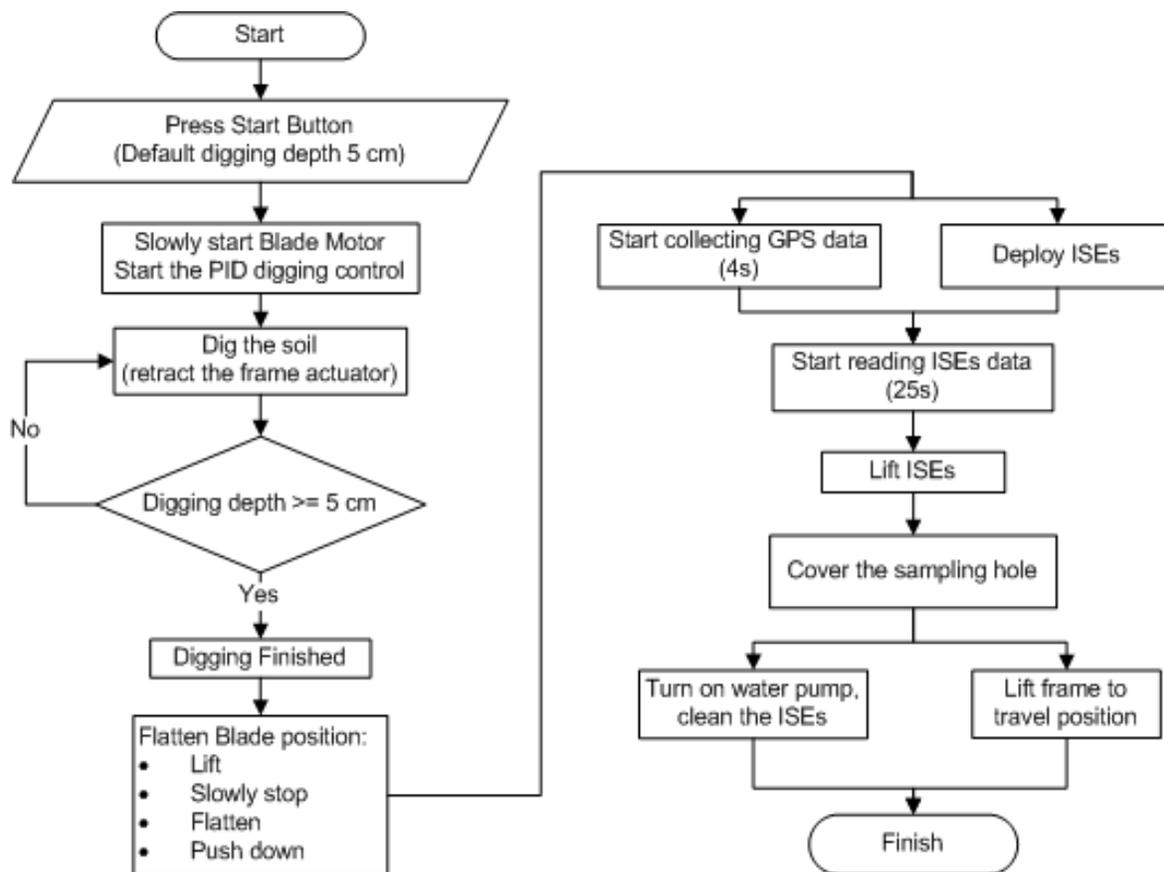


Figure 3-1. The OSA operation flow

3.2.2. Physical Design

SolidWorks (Dassault Systemes S.A., Waltham, Massachusetts, USA) was used to design the optimized OSA prototype. The new design (Figure 3-2) included two sets of parallel linkages which were used to accommodate various depth measurements and to reduce the digging curvature (Figure 3-3), platform vibration and sampling footprint as compared to the original OSA system. Additionally, instead of a direct drive, a sprocket-chain mechanism was utilized to drive the rotating blades. This method was selected to minimize electrical noise from the digging blade motor and provide more ground clearance; this resulted in improved digging depth and provided protection from ground stubble. A series of bracketry were designed to accommodate a linear actuator to restore the soil after each measurement. Furthermore, instead of two car windshield washer nozzles, three TeeJet XR11002VK nozzles (TeeJet Technologies, Glendale Heights, Illinois, USA) were used to provide better water spray which is necessary for cleaning the ISEs.

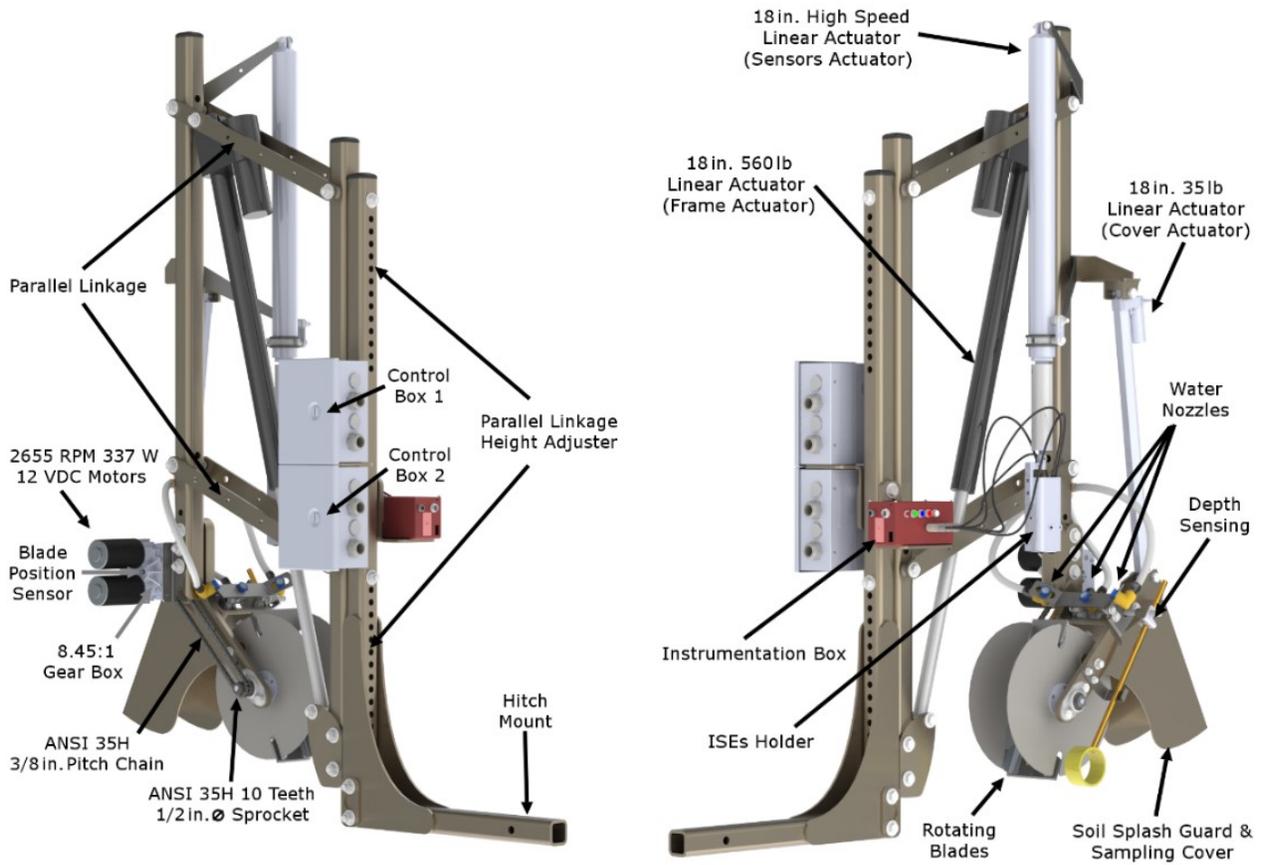


Figure 3-2. The new OSA design

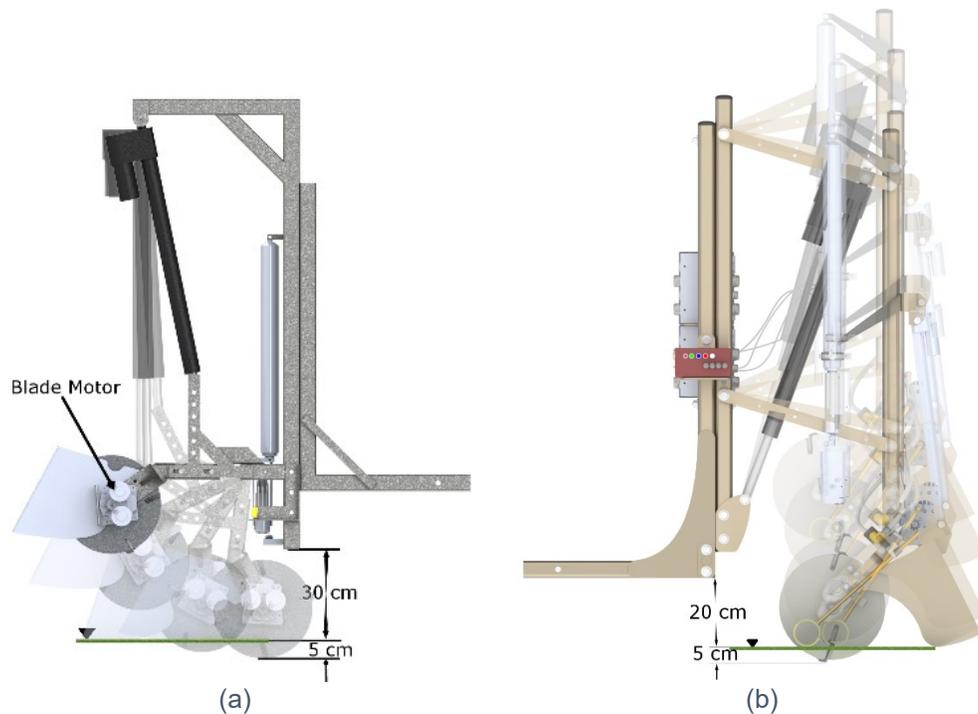


Figure 3-3. OSA digging curvature comparison.
 (a) the old and (b) the new design

3.2.3. Electronics Components

The detailed OSA electronic layout is presented in Figure 3-4, while the schematic wiring diagram is presented in Appendix 9.1. Two Arduino Uno microcontrollers were used to control various motors using a pulse width modulation (PWM) signal as well as for acquiring feedback from the various sensors. The first Arduino was responsible for controlling the digging action and blade position, while the second Arduino was responsible for deploying the ISEs and covering the sample hole. Each Arduino was enclosed in a dedicated metallic enclosure. A dual 12 VDC motor (2655 rpm, 337 W) with 8.45:1 ToughBox Mini gear box (AndyMark Inc., Kokomo, Indiana, USA) rotated the digging blades. During the field operation, the blades were set to always rotate at full speed. The load of the blades was controlled by the movement of a 45 cm linear actuator (Robotzone LLC, Winfield, Kansas, USA) which moves the frame accordingly. After the digging was completed, a 45 cm high speed linear actuator (Progressive Automations Inc., Richmond, British Columbia, Canada) was used to deploy the ISE sensors. The ISE measurement was set for 25 s which was more than sufficient to obtain full sensor response and steady-state behavior (Adsett et al., 1999; Adamchuk et al., 1999).

Subsequently, a 5 to 12 VDC relay (Velleman Inc., Fort Worth, Texas, USA) was used to activate the Simer Blue Water self-priming pump (Pentair Plc., Delavan, Wisconsin, USA) to pump water into the 9.5 mm tube line for washing the ISEs. To finish the measurement cycle, another 45 cm linear actuator (Progressive Automations Inc., Richmond, British Columbia, Canada) was used to move the soil debris guard to restore the soil into the sampling hole. All the motors and actuators were each controlled using the 900 W Talon SR servo motor controller (Cross the Road Electronics LLC, Macomb, Michigan, USA).

The new OSA was also equipped with various sensors to achieve faster digging time and more efficient energy utilization. Each blade motor load was monitored using 50 A hall effect ACS758 current sensors (DFROBOT, Pudong, Shanghai, China). After the digging was completed, the blades needed to be positioned in such a way as to not obstruct the deployment of the sensors. To monitor the blade positions, a 200 k Ω continuous rotation potentiometer (Precision Sales and Equipment Inc., Newtown Square, Pennsylvania, USA) was placed at the blade motor output shaft. To measure the digging depth, an AN1 120° linear rotating potentiometer (ZF Electronic Systems LLC, Pleasant Prairie, Wisconsin, USA) was used. Additionally, all actuators were equipped with 50 A hall effect ACS758 current sensors (DFROBOT, Pudong, Shanghai, China) to monitor their load as well as a safety measure.

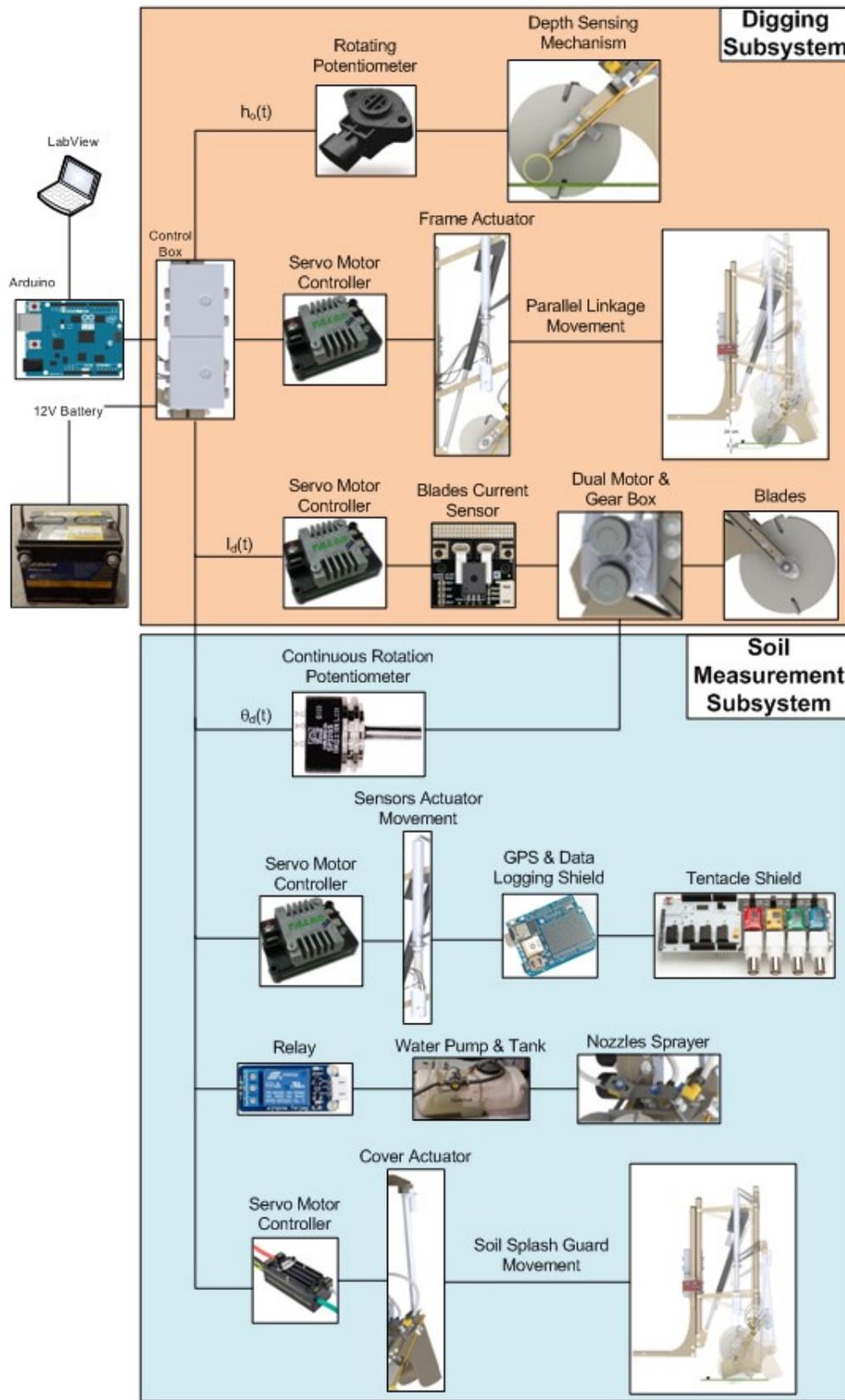


Figure 3-4. The detailed OSA electrical layout

3.2.4. OSA Data Acquisition System (DAQ) and Graphical User Interface (GUI)

The original OSA platform used expensive NI-MyDAQ and NI-USB-6216 data acquisition cards (National Instruments, Austin, Texas, USA). In the new OSA design, an Arduino Uno was used as a data acquisition system (DAQ) dedicated primarily for ISE measurements. The Arduino DAQ was comprised of Whitebox Labs' Tentacle shield (Meister Whiteboxes GmbH, Basel, Switzerland) equipped with four Atlas Scientific ORP (oxidation-reduction potential) circuits (Atlas Scientific LLC, Brooklyn, New York, USA) each with 1 sample/s sampling rate. This setup allows simultaneous ISEs data recording. Additionally, a GPS data logger and Bluetooth shield were added to the DAQ to provide georeferenced data and to ease mobile connectivity. The DAQ schematic diagram is represented in Appendix 9.1. A GUI was developed using LabVIEW 2015 (National Instruments Corp., Austin, Texas, USA) to ease the interface among all of the Arduinos used for controlling the motors and actuators, and also for data acquisition (Figure 3-5).

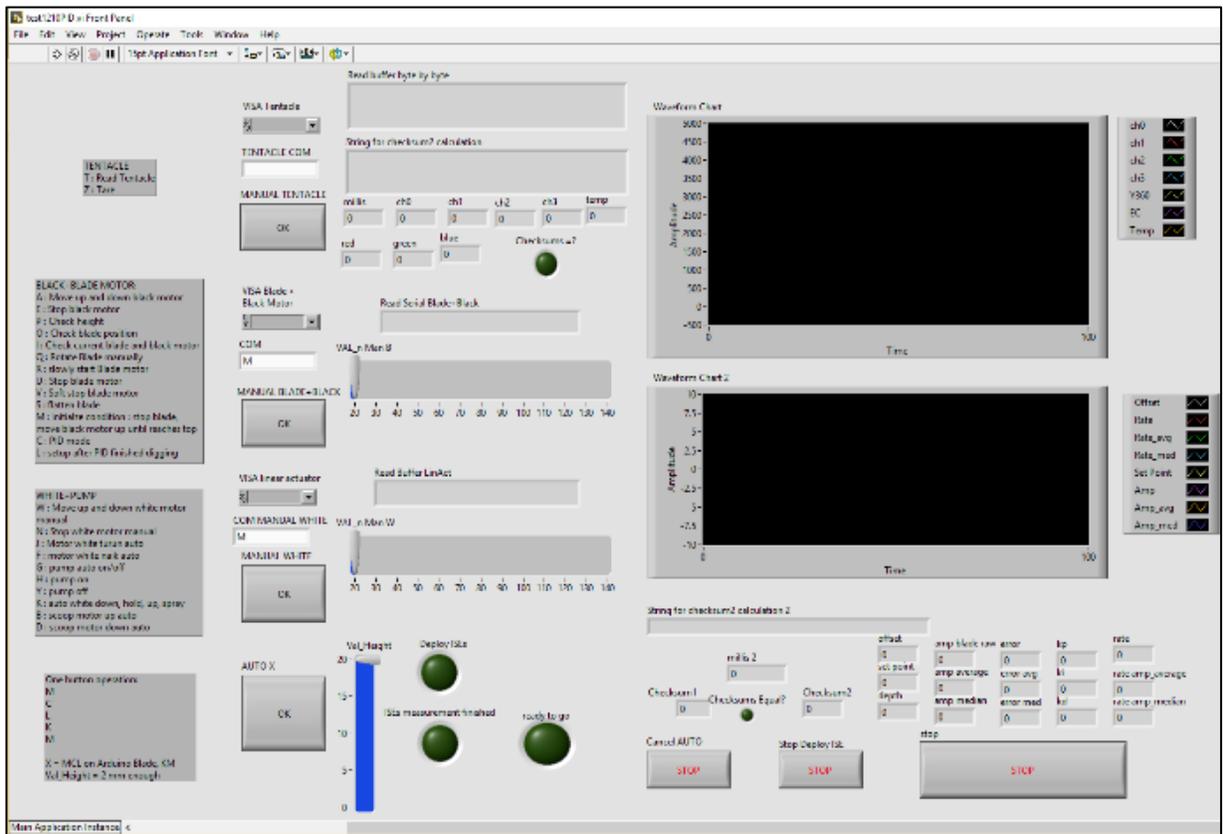


Figure 3-5. LabVIEW OSA GUI

3.2.5. OSA Control Systems

The main focus of the OSA control design development was to achieve an optimum digging time under various field conditions. The OSA digging control should dig the soil as fast as possible, stop when the desired digging depth was reached while at the same time, has a rapid response in adjusting the digging load by retracting or lowering the parallel linkage according to the field condition (e.g. soil surface cover, compaction, etc.). The OSA digging controller should maintain the blade motor load at the digging current set point and not exceeding the breaker current.

Measurement cycle of on-the-go soil sensing platform for soil chemical properties using ISE or ISFET depends on the sensor's performance (stabilization and response time). Adamchuk et al. (2007) found that using Veris Mobile Sensor Platform (Veris Technologies, Inc. Salina, Kansas, USA), the average measurement cycle was 10 s. Soil sensing system developed by Viscarra Rossel et al. (2005) achieved a total time of 27 s with 3 s spent on soil sampling. The OSA digging controller should response promptly to compensate for the time taken for the OSA to stop when conducting DSM using ISEs. Therefore, the OSA controller objectives were to produce the desired transient response of less than 300 ms with reduced steady-state error and to attain controller stability. The expected digging time was 3 s for ideal field condition and 15 s for compacted soil with vegetation cover.

Functional Block Diagram

To ease the OSA control development process, a series of functional block diagrams were needed. The two most important block diagrams were the digging action (Figure 3-6) and digging depth block diagram (Figure 3-7) which represent the feedback system configuration. Initially, the OSA operator needs to input the desired digging depth (digging depth setpoint, h_{sp}) via the OSA GUI. From the controller tuning, the load limit (digging current setpoint, I_{sp}) was agreed to be 5 A. Then, the Arduino can accelerate the blade motors until it reaches the maximum rotational speed. Subsequently, while performing the digging action, the current of the blade motors was measured and used as feedback control information because this current provides real-time digging load data. The load of the blade motor is comprised of two components: the static load from rotating the blades (no-load current) and the dynamic load from the soil mechanical resistance. Finally, the

movement of the frame actuator was controlled based on the digging load feedback information obtained from the blade motor current sensors. After the desired digging depth was achieved, the frame linear actuator movement was stopped slowly to avoid a spike in the current.

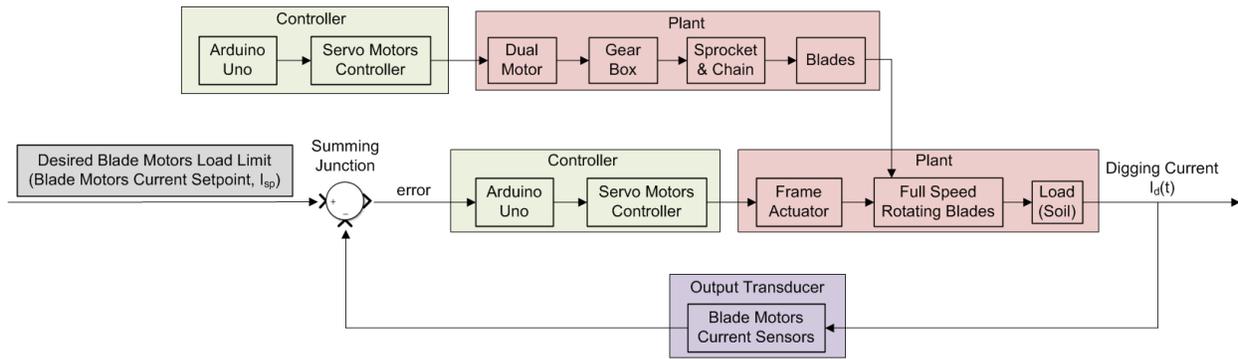


Figure 3-6. The OSA digging block diagram

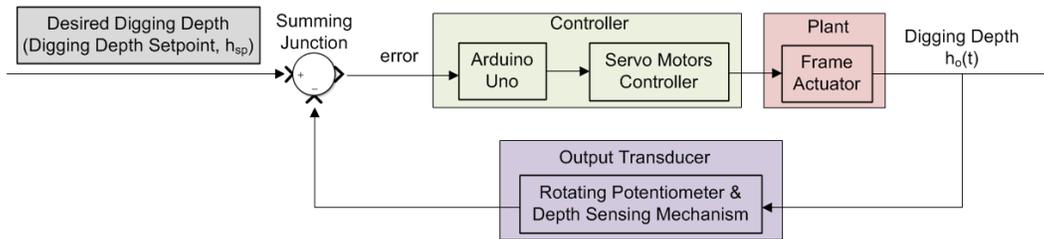


Figure 3-7. The OSA digging depth block diagram

After stopping the frame linear actuator, the Arduino will slowly stop the blade rotation to avoid a spike in the current. Subsequently, the blades need to be positioned horizontally such that they do not interfere with deploying the ISE sensors. To accomplish this task, another feedback system was constructed (Figure 3-8). The closed-loop flatten blades system configuration consists of a continuous rotation potentiometer placed at the blade motor output shaft to detect the blade position. When the blades are stopped after the digging is finished, the Arduino will rotate the blades slowly and stop the blade when the position is in the blades flatness setpoint range (θ_{sp}). The flatness setpoint was determined to have 30 to 45° range before reaching the horizontal position (0°) to accommodate the sprocket-chain slack, gear box backlash and motor coasting when the blades stopped.

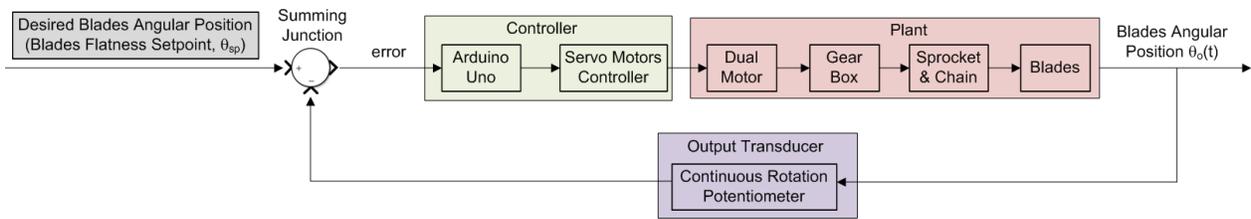


Figure 3-8. The OSA flatten blades block diagram

The soil measurement can be started after the blades are in the flat position. The 45 cm high speed linear actuator will be extended to deploy the ISEs sensor to touch the prepared soil. While deploying the ISEs, the DAQ starts the GPS data acquisition for 5 s and by the time the sensors reach the soil surface, the soil chemical measurement can be started. Total time to measure the soil was determined to be not more than 20 s as this was to be typical for the DSM method. After the soil measurement, the sensor actuator retracts to restore the ISEs to its original position. As the sensor linear actuator is equipped with position limit switches which will cut the power if the actuator reaches its maximum extension or minimum retraction position. Therefore, there is no need to include position feedback or a current sensor into the system. Consequently, the sensor deployment block diagram was configured as an open-loop system (Figure 3-9).

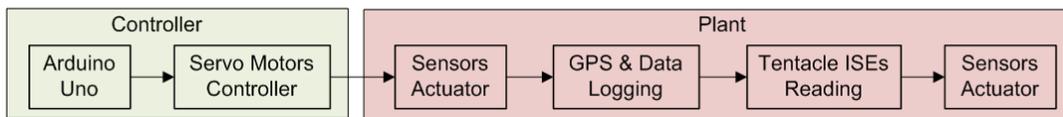


Figure 3-9. The OSA sensor deployment block diagram

Sensor washing and covering the sampling hole were subsequent operations after the sensor actuator returned the ISEs holder to its original position. These operations were based on the timing control after the sensor deployment operation finished. Therefore, they were configured as an open-loop system (Figure 3-10). The washing procedure started by engaging the water pump relay to close the battery-pump circuit. Water was pumped into the 3/8 in. tube line and three nozzles to provide the fine spray of water necessary for cleaning the ISEs. The washing period lasted for 3 s which was sufficient to clean four ISEs.

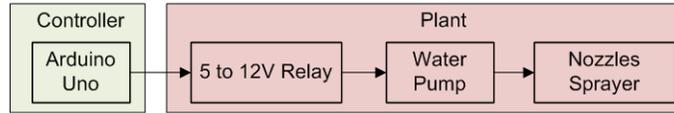


Figure 3-10. The OSA washing block diagram

Covering the sampling hole was done by moving the OSA soil debris guard using a linear actuator (Figure 3-11). The soil sample covering action was time triggered. While the soil debris guard actuator was moving, the frame linear actuator also needed to move the frame upward to provide a flat soil covering. After this was finished, the frame actuator extended to its longest stroke to lift the frame to the transport position.

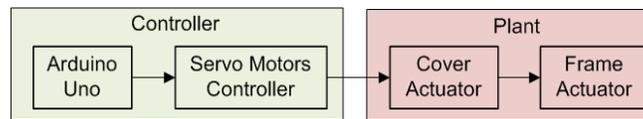


Figure 3-11. The OSA soil covering block diagram

White-Box Modelling

To design a control system, a white box approach was first employed to model the OSA digging process. The white-box controller modelling was used to give a general overview on the expected OSA digging response. This method provides valuable information about the appropriate controller design and its parameters.

In the white-box modelling, the controller's plant consists of a frame linear actuator, linear actuator motor controller, blade motor and soil (Figure 3-6). Electronic components were modelled based on the manufacturer specification information. Matlab R2015b (MathWorks Inc. Natick, Massachusetts, USA) Simulink was used to model the OSA digging controller.

Frame Linear Actuator Motor Modelling

The frame linear actuator consists of a DC motor, gear train and leadscrew. In the Matlab Simulink library, a DC motor can be parameterized using several options: by equivalent circuit parameters, by stall torque and no-load speed, or by rated power, rated speed and no-load speed. The 45 cm frame linear actuator has a mass of 5.7 kg, lead screw pitch (ϕ_{ls}) of 5 mm, gear ratio (G_r) of 1:5, no-load (v_{nl}) and rated linear speed (v_r) of 67.1 and 47.2 mm/s, respectively. The actuator rated load is 2500 N with no-load (I_{nl}) and

rated current (I_r) of 3.4 and 26.4 A, respectively. From this information, it was convenient to use the latest parameterization option. The following equations are needed to determine the additional frame actuator model parameter:

$$v_m = \frac{v_{nl}}{\phi_{ls}} \times G_r \times 60 \quad (3-1)$$

$$v_{mr} = \frac{v_{rl}}{\phi_{ls}} \times G_r \times 60 \quad (3-2)$$

$$P_{mr} = F_r \times \frac{v_{rl}}{1000} \quad (3-3)$$

where v_m is the DC motor no-load speed (RPM), v_{mr} is the rated motor speed (RPM) and P_{mr} is the rated power (W).

The rotor damping parameterization was set by no-load current instead of by the damping value because it requires already known information of no-load current and DC supply voltage. Several assumptions need to be made to fulfill the model parametrization such as: armature inductance and the OSA moving frame mass to calculate the rotor inertia. The armature inductance was assumed using Matlab's default value of 12 μ H. The linear actuator DC motor case has a 63.5 mm diameter, thus, the rotor diameter was assumed to be 45 mm. The total weight of the motor was 60% of the product mass and the rotor mass was 20% of the motor weight, which resulted in 0.68 kg. Thus, the rotor inertia (J) (assuming it was a solid cylinder) was 1.7×10^{-4} kg m².

A force signal opposite to the linear actuator extension direction was used to test the model for its output. The test force starts with 0 N as the no-load value and it is stepped up to 2500 N as rated load. The result confirms the model's linear no-load lead screw speed, motor rotational speed, motor no-load current and in-rush current settling time of 200 ms. However, it underestimates the no-load in-rush current and rated current. To obtain the appropriate in-rush current, the armature inductance was set to 4 mH, which satisfies the test in-rush current of 34 A. The simulation reveals that the rated current was 17.4 A, or 9.0 A below the specified rated current. This result might be attributed to our idealized no friction model assumption. Nonetheless, the model performs well for

estimating linear lead screw speed which was directly related to digging load. Furthermore, by adding more load, the maximum current can be estimated to be as much as 51 A at 8.5 kN (Figure 3-12).

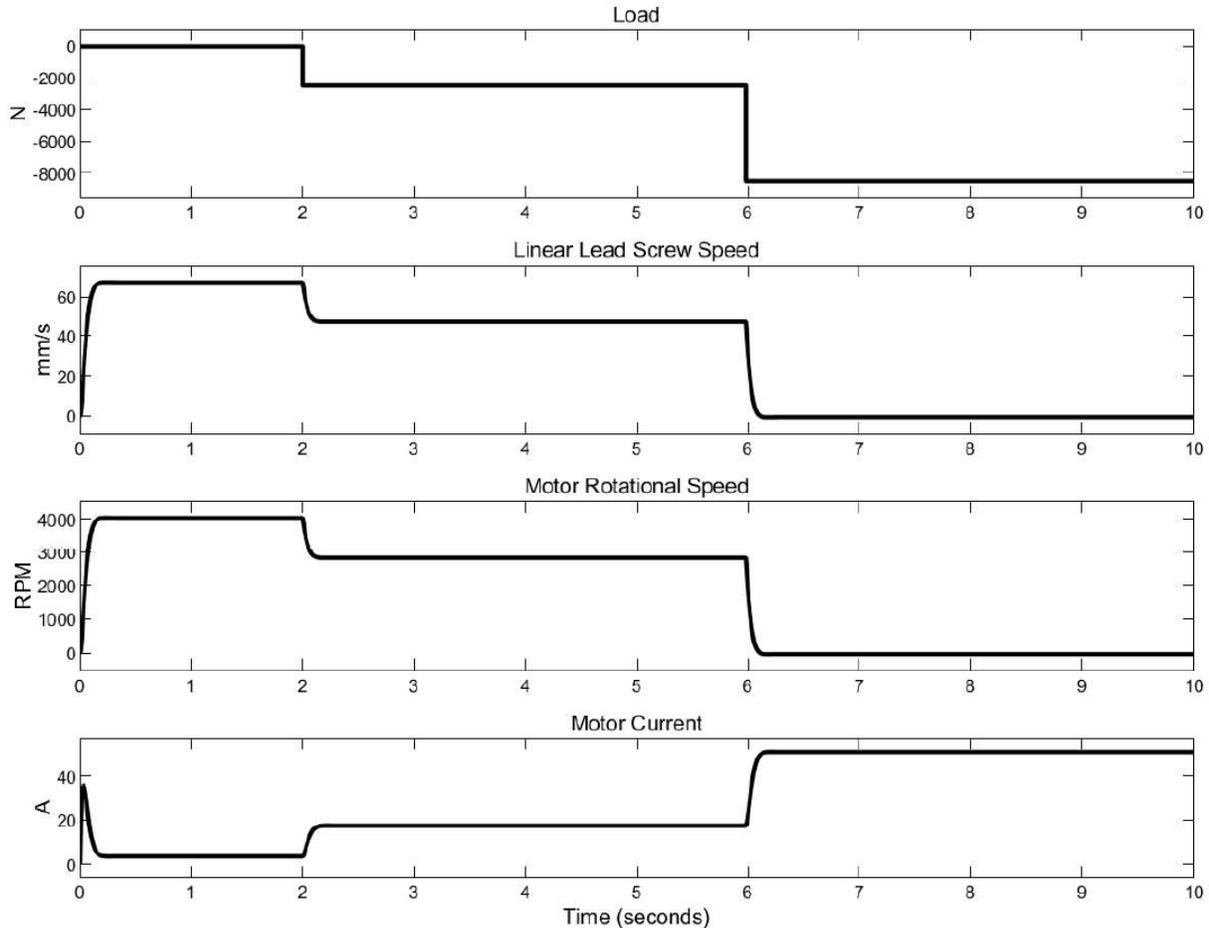


Figure 3-12. Frame linear actuator model response

Linear Actuator Motor Controller Modelling

For the actuator speed controller, an Arduino and 900 W Talon SR servo motor was used. These combinations work based on the Arduino PWM servo library. The Arduino was used to produce 50 Hz PWM servo signal using the `servo.write()` command. Then, the PWM signal was converted by the Talon SR motor controller to voltage to move the linear actuator motor. The Talon SR motor controller has an H-bridge configuration that adjusts the speed and direction of the motor. Therefore, to model the motor controller, a controlled PWM voltage and H-bridge Simulink blocks were used.

The controlled PWM voltage block was set to generate PWM with a frequency of 50 Hz and 5 V output voltage based on the Arduino servo library. Based on a bench test, the linear actuator's DC motor will stop at `servo.write(95)` or equivalent to 1.5 ms PWM pulse. The maximum retraction and extension speed reached at 0.5 and 2.5 ms pulse or equivalent with 0 and 180 `servo.write()` command, respectively. Since the digging action happens when the linear actuator is in the retraction movement (lowering the OSA frame), the PWM input was scaled to 95 as 0% duty cycle and 0 as 100% duty cycle. However, since the controlled PWM voltage block cannot handle the inverse duty cycle, we needed to reverse the input PWM feed to 95 as 0% duty cycle and 180 as 100% duty cycle. Furthermore, since the digging occurs when the linear actuator is retracted, we also needed to reverse the screw rotation to be left-handed.

The Simulink model of the OSA frame linear actuator can be seen in Figure 3-13. In this model, an idealized frictionless motor and lead screw were used. Consequently, we must put a static zero weight acting on the leadscrew. Otherwise, the lead screw will move (depending on the direction of the force) while the PWM signal is 95. The Simulink H-bridge block was set to have 5 and 12 V PWM input and output voltage amplitude, respectively. A PWM step signal of 95, 70, 0, 50 and 95 were employed to test the motor control model response. Initially, the full motor speed was below specification. A modification to the H-bridge total resistance to a value of 1 m Ω was found to produce an appropriate motor response. The negative lead screw speed and rod position means the actuator was retracting and consequently, providing deeper digging (Figure 3-14).

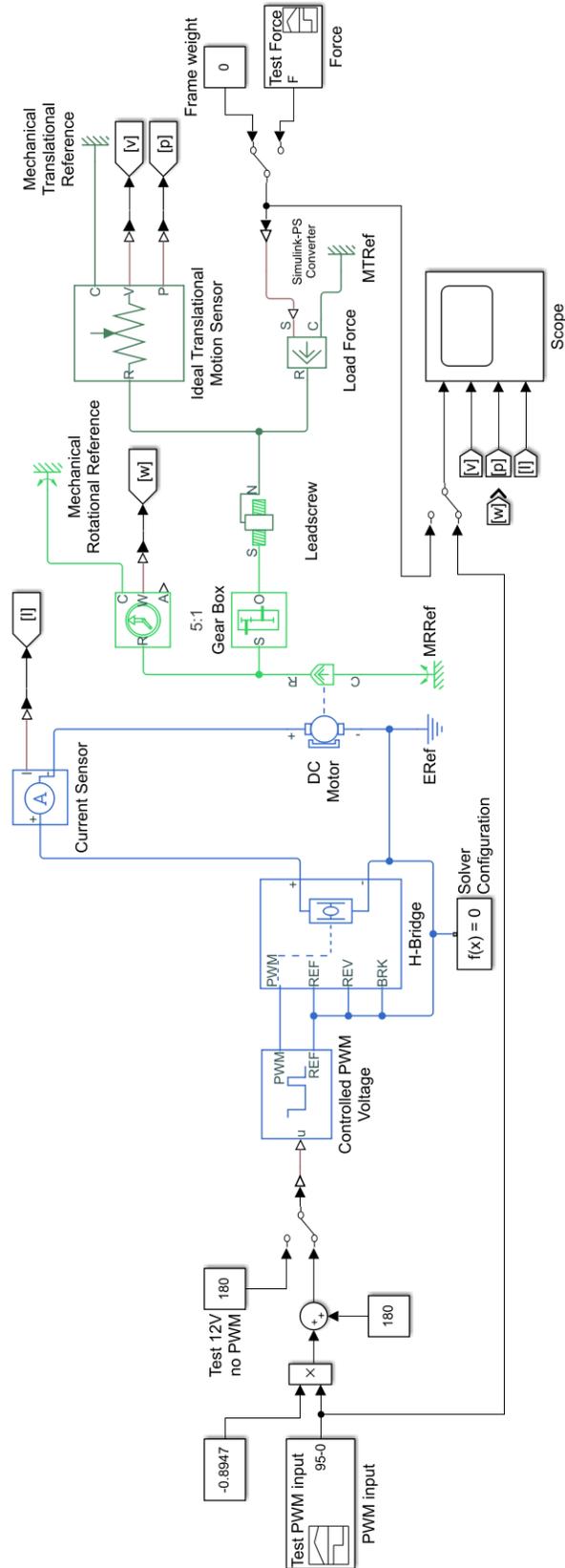


Figure 3-13. Linear actuator Simulink model

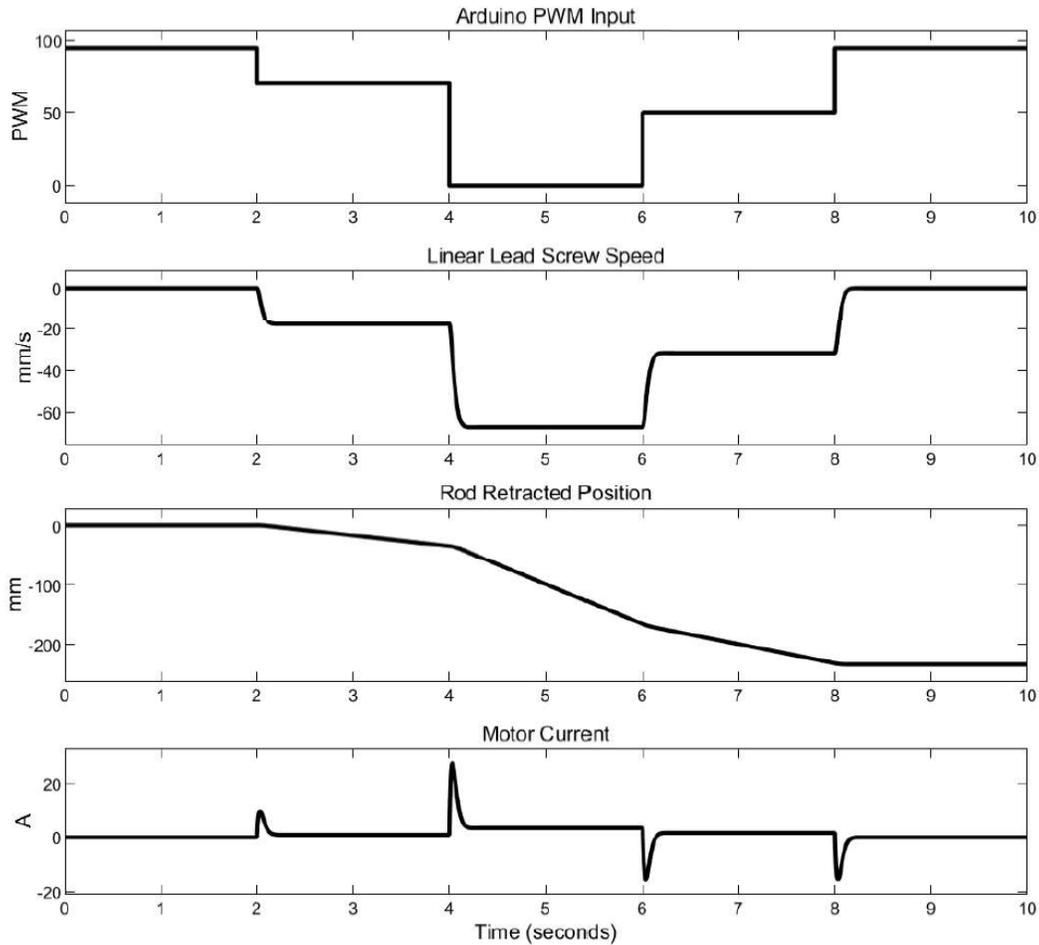


Figure 3-14. Frame linear actuator PWM motor controller response

Blade Motor Modelling

The blade motor has a mass of 1.27 kg, shaft radius (r_s) of 4 mm, no-load (v_{nl}) and rated speed (v_r) of 5310 and 2655 RPM, respectively. The no-load (I_{nl}), rated (I_r) and stall current (I_s) are 2.5, 66.9 and 131.2 A, respectively. Whereas the rated (T_r) and stall torque (T_s) are 1.21 and 2.42 Nm, respectively. The motor has 91 mΩ armature resistance, rated power of 336.5 W and torque constant of 1.88×10^{-2} Nm/A. With those specification, it was convenient to model the blade motor using the stall torque and no-load speed option in the Simulink DC motor block.

Additional calculations were needed to calculate the rotor inertia. The rotor mass was assumed to be 50% of the motor mass. With 25.4 mm rotor radius, the rotor inertia was found to be 2.1×10^{-4} kg m². To test the model, a force of 305 and 609 N was fed as rated

and stall load, respectively, in the direction opposite of the blade rotation (negative force). The simulation result agrees with the specification motor RPM. However, for the rated and stall load, the motor current underestimates the specification by 8.3 and 16.5 A, respectively. This underestimation was quite severe since the blade motor current reading will be used as feedback for the digging controller design. As a result, the DC motor parameterization option needed to be changed into “by equivalent circuit”. This option provides armature resistance and torque constant to the model which is beneficial to compensate for the motor current underestimation. Although it produces a slight overestimation in motor speed, the model was producing a proper motor current of 2.5, 65.7, and 128.8 A for no-load, rated load and stall load, respectively (Figure 3-15). To obtain a suitable in-rush current response, the armature inductance was set to 1.5 mH.

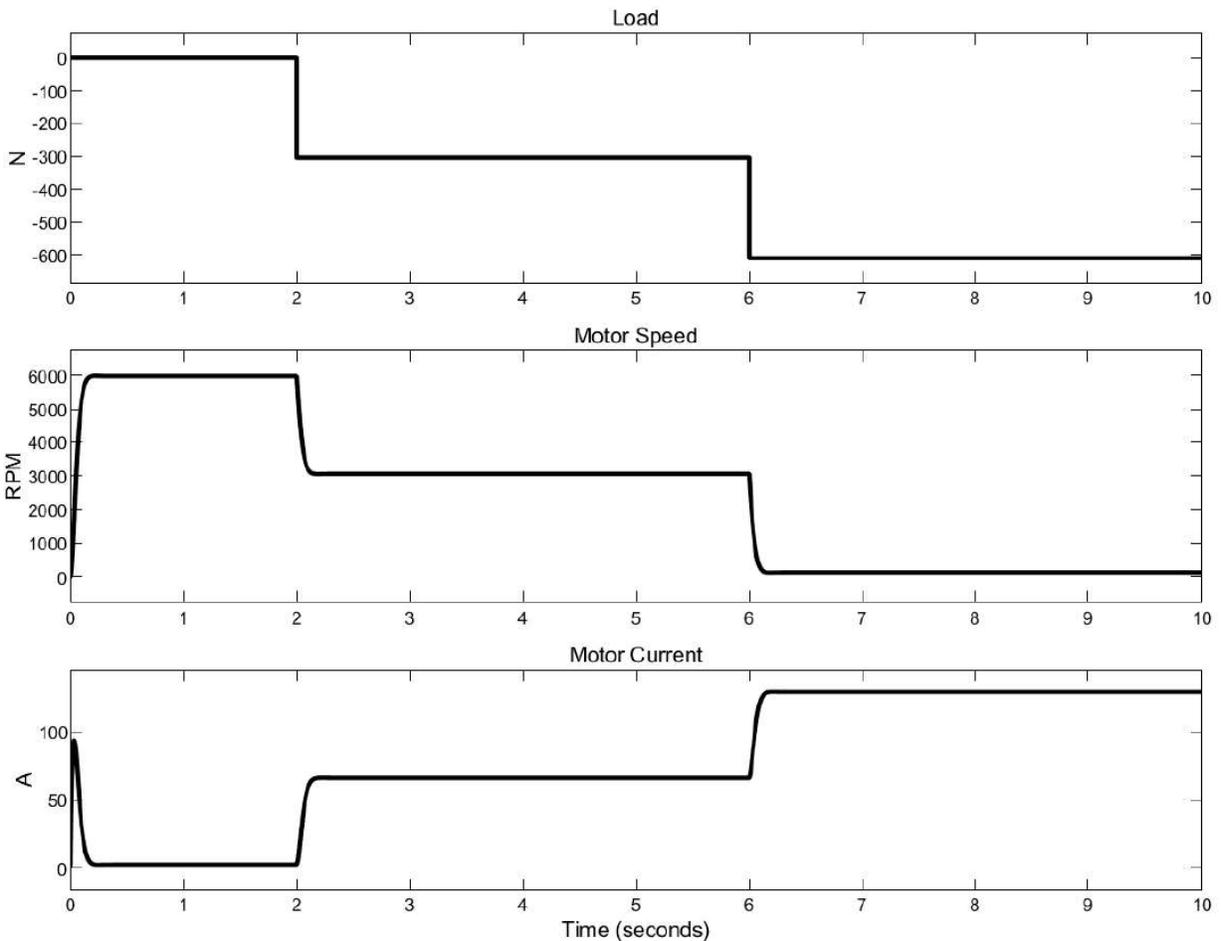


Figure 3-15. Blade motor simulation response after adjustment

Soil Resistance Modelling

Reece (1964) developed a universal earthmoving equation which can be used to estimate the soil resistance for the OSA blade load input:

$$P = (\gamma g d^2 N_\gamma + c d N_c + q d N_q) w \quad (3-4)$$

where P is total tool force (N); γ is total soil density (kg m^{-3}); g is gravitational acceleration (m s^{-2}); d is tool working depth (m); c is soil cohesion strength (Pa); q is vertical surcharge pressure acting on soil surface (Pa); w is tool width (m); and N_γ , N_c and N_q are dimensionless factor which are a function of soil angle of internal friction (ϕ°).

This equation was developed for two-dimensional problems, such as tillage blade which is similar to the OSA blade design. The blade was assumed to be smooth and vertical, meaning that there was no upwards shear stress acting on the blade. Thus, the total force to move the soil was perpendicular to the blade. Since the OSA was designed to sample a field where there is no surcharge pressure acting on the soil surface, equation (3-4) can be simplified to:

$$P = (\gamma g d^2 N_\gamma + c d N_c) w \quad (3-5)$$

The OSA blade is a horizontal rotating blade, the maximum load happens when the blade first contacts the soil as it will compress the soil until the blade turns vertical. The blade was assumed to work on a sandy loam soil with cohesion strength of 1.4 kPa and density of 1.4 g cm^{-3} . For sandy loam with $\phi = 28^\circ$, the N_γ and N_c values are 2 and 4, respectively (McKyes, 1985). With 120 mm OSA blade width, then the total tool force is dependent on the blade working depth, which is a function of blade rotation speed and OSA frame linear actuator speed. To calculate the blade working depth, the following equation was used:

$$d_b = \frac{v_{la}}{v_b} \times 60 \quad (3-6)$$

where d_b is blade working depth (mm), v_{la} is the linear actuator speed (mm s^{-1}) and v_b is the blade rotation speed (RPM).

Since the linear actuator has 20° angle from blade frame (Figure 3-16), the retraction actuator speed needs to be multiplied with cosine angle to get the blade frame downward speed. Moreover, as there are two blades in the blade drum, the blade depth (d) would be half of that calculated previously, thus equation (3-6) becomes:

$$d = \frac{1}{2} \frac{v_{la} \cos(20^\circ)}{v_b} \times 60 \quad (3-7)$$

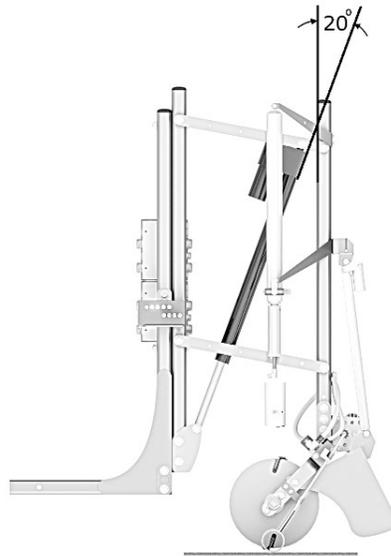


Figure 3-16. The angle between linear actuator and blade frame when digging

The complete Simulink white-box controller model which consists of a frame linear actuator, linear actuator motor controller, blade motor and soil is presented in Figure 3-17.

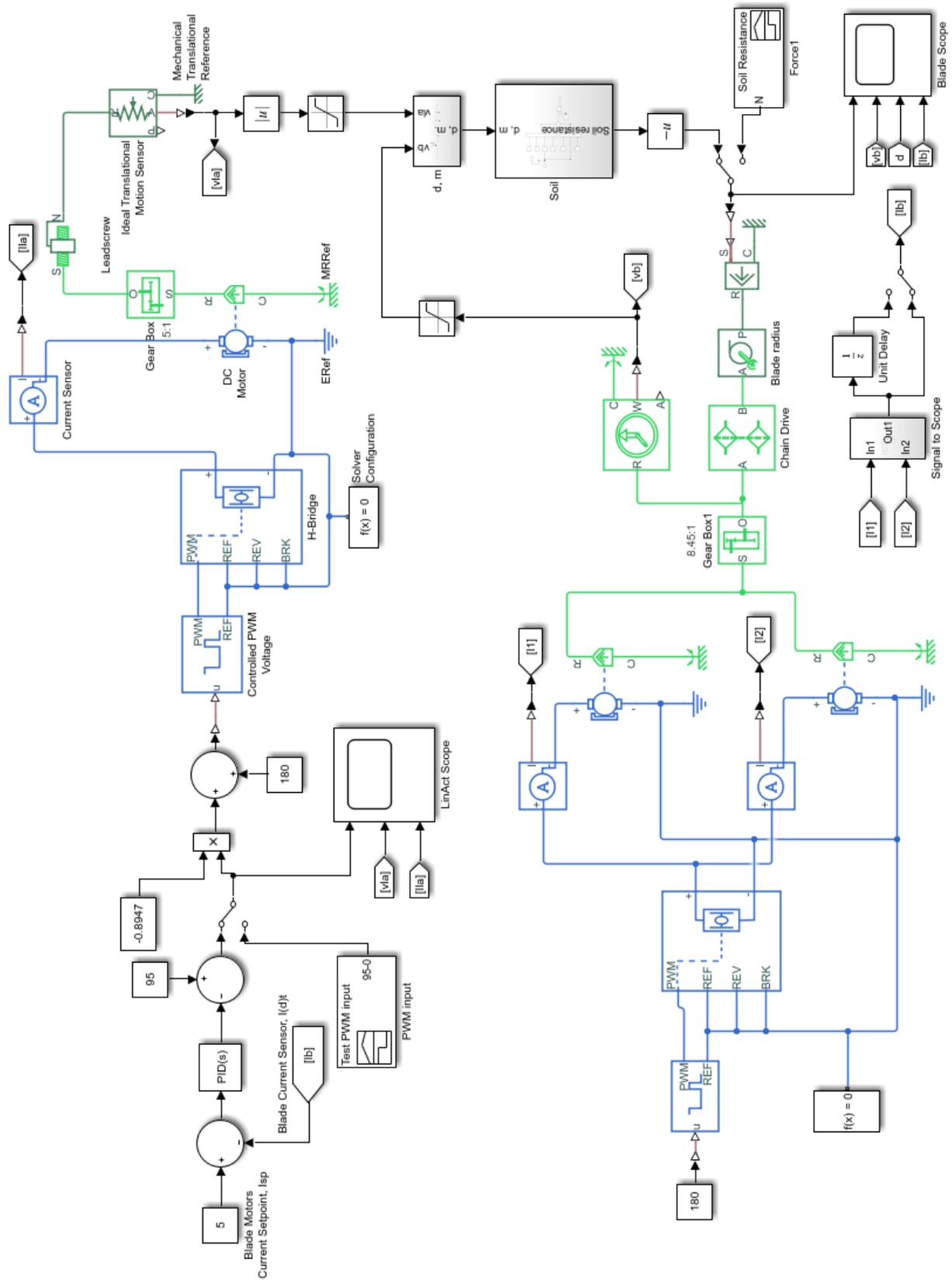


Figure 3-17. White-box OSA digging controller model

Black-Box Modelling

In the black-box modelling, the digging controller's plant which consists of the frame actuator, rotating blade and soil are lumped together. To get the plant model, a step PWM signal was introduced to the frame actuator so the OSA blade would start to perform the digging action. The digging current was recorded and parameterized. The blade current sensor reading needs to be filtered to reduce noise. A median filter of 5 samples with an average sampling time of 92 ms was sufficient to filter the outlier and to provide a decent response (Figure 3-18).

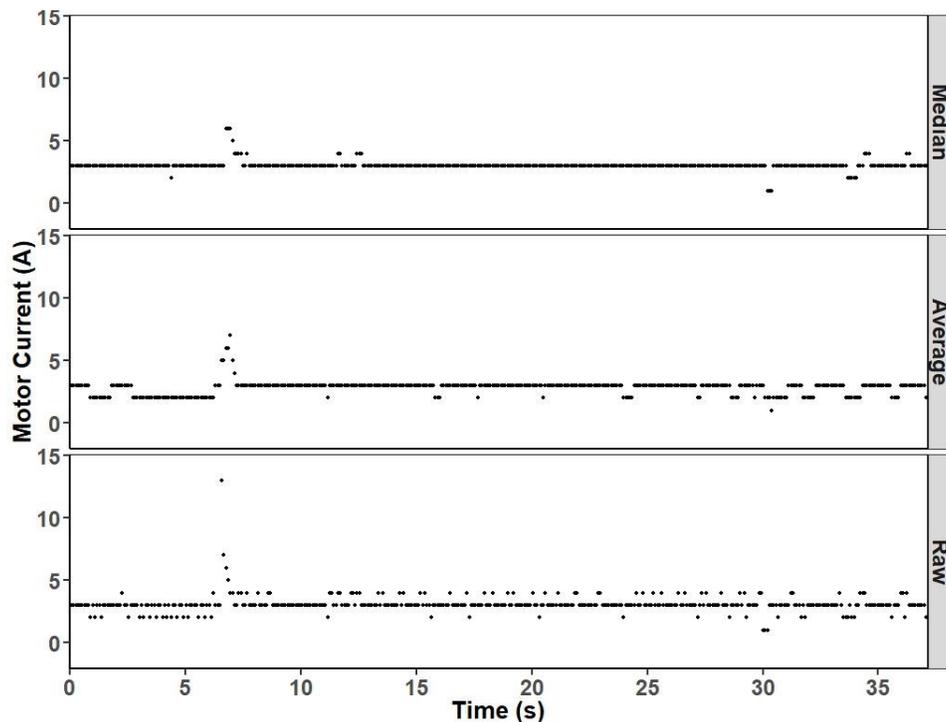


Figure 3-18. Comparison of 5 points average and median filter

Step PWM signal of 77 or equal to 28 and 26.3 mm s⁻¹ linear actuator and blade frame downward speed, respectively, were chosen as the input signal. The field test was conducted for 500 ms to observe the response of the controller's plant. The conservative step PWM signal and short test time were useful since the OSA has a high momentum and inertia during digging from its high-speed rotating blade and frame movement. Therefore, it is important to see the initial response when the blade starts hitting the soil for the first time to better represent the controller's plant.

There are two parameterization methods used in this black-box modelling, namely, underdamped second-order system and Ziegler-Nichols. The first-order system was not considered because when the blade initially hit the soil, it is expected to produce a current spike. Therefore, it does not represent the first-order system, instead it is more towards the underdamped second-order system.

Underdamped Second-Order System

From the field test data (Figure 3-19), the plant shows a similarity with the second-order underdamped system where there was a current overshoot when the blade initially contacted the soil, without oscillating. Then, the current converges to a steady state value. To ease the plant identification transfer function, a Matlab GUI (Figure 3-20) was developed. The GUI also provided manual ω_n^2 and $2\zeta\omega_n$ adjustments to fulfill the designer's needs.

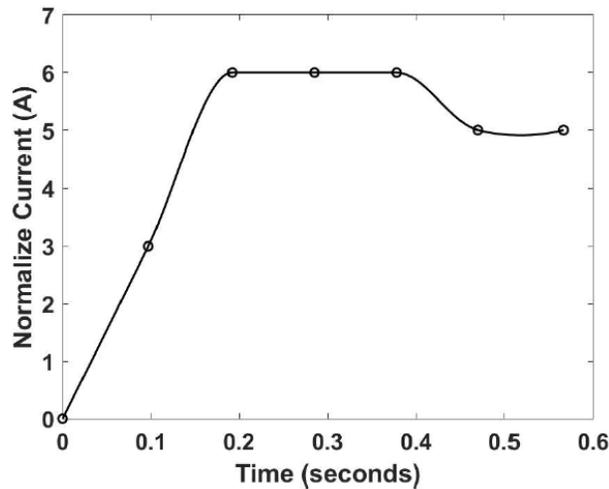


Figure 3-19. OSA blade current response

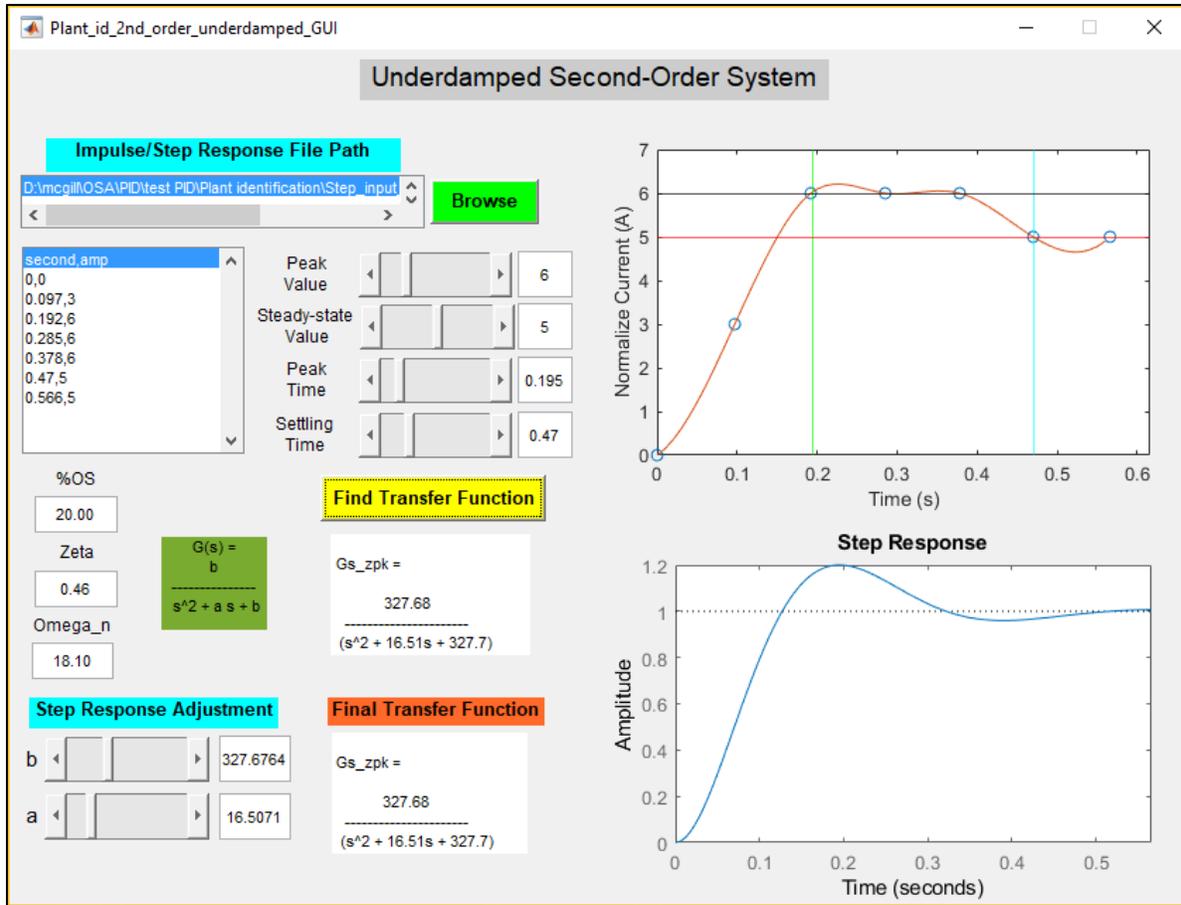


Figure 3-20. OSA plant second-order transfer function identification GUI

The general second-order transfer function is given below:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3-8)$$

where ω_n and ζ are the system's natural frequency and damping ratio, respectively. In order to use (3-8), several second-order transfer function specifications need to be identified, namely: peak time (T_p , the time needed to reach maximum response value), settling time (T_s , the time needed to reach within $\pm 2\%$ steady state value) and percent overshoot (%OS, the percentage of peak value from the steady state value).

From the experimental result, T_p , T_s and %OS were found to be 0.195 s, 0.47 s and 20% respectively. With these specifications known, ζ and ω_n can be calculated using the following equations:

$$\zeta = \frac{-\ln\left(\frac{\%OS}{100}\right)}{\sqrt{\pi^2 + \ln^2\left(\frac{\%OS}{100}\right)}} \quad (3-9)$$

$$\omega_n = \frac{\pi}{T_P\sqrt{1-\zeta}} \quad (3-10)$$

Thus, the OSA digging subsystem transfer function is:

$$G(s) = \frac{327.7}{s^2 + 16.51s + 327.7} \quad (3-11)$$

and the equivalent closed-loop transfer function is given by:

$$T(s) = \frac{327.7}{s^2 + 16.51s + 655.4} \quad (3-12)$$

Before considering the appropriate controller design, a Routh-Hurwitz criterion was used to identify the plant's stability (Nise, 2015). From the Routh-Hurwitz analysis, there is no pole at the Laplace transform s-domain right half-plane. Thus, the system was considered stable with poles $-8.25 \pm 24.23i$.

Ziegler-Nichols Method

Another field test response can be seen in Figure 3-21. The Ziegler-Nichols method can be used to characterize the controller's parameters from this S-shaped curve (Ogata, 2010). From Figure 3-21, we can obtain the delay time (L) and time constant (T) of 0.15 and 0.14 s, respectively. The selection of different controller parameters is summarized in Table 3-1.

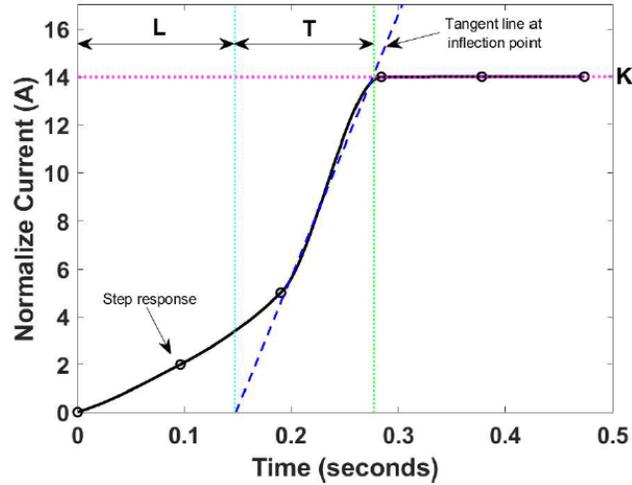


Figure 3-21. S-shaped curve controller's field test response

Table 3-1. Ziegler-Nichols rules and result

Type of Controller	K_p		T_i		T_d	
	Rule	Value	Rule	Value	Rule	Value
P	$\frac{T}{L}$	0.93	∞	∞	0	0
PI	$0.9 \frac{T}{L}$	0.84	$\frac{T}{0.3}$	0.47	0	0
PID	$1.2 \frac{T}{L}$	1.12	$2L$	0.3	$0.5L$	0.075

OSA Digging Controller Development

A proportional-integral-derivative (PID) controller was selected to account for the effect of friction, system's inertia and external load variation (Sariyildiz et al., 2015). The goal of the PID controller was to provide an appropriate PWM rate input for the linear actuator to maintain a desirable digging speed under different field condition and reducing the controller's response lag due to the OSA inertia while digging. The controller should keep the linear actuator moving the frame down (PWM < 95) to continue digging as long as the blade current (I_d) was less than the digging current setpoint (I_{sp}). Conversely, it should stop digging (PWM > 95) when the I_d exceeds I_{sp} .

For the controller simulation, the digging current setpoint was set to be 5 A or 2.5 A more than the no-load blade current. Additionally, a 95 offset was introduced to the controller as 0% duty cycle to makes the controller PWM output remain at 95 (no digging)

when all the controller parameters are at zero. The formula for the white-box PID controller simulation is presented below:

$$PWM\ rate = 95 - \left(P + I \frac{1}{s} + D \frac{N}{1 + N \frac{1}{s}} \right) \quad (3-13)$$

where P, I and D are proportional, integral and derivative control parameter, respectively and N is the simulation low pass filter coefficient which filter the electrical and mechanical noise expected from the controller's plant model.

Using the Matlab's PID tuner functionality, the white-box PID controller parameters were found to be 140.4, 1473 and 3.1 for the proportional, integral and derivative parameter, respectively with simulation filter coefficient of 1915. The simulated controller response has 310 ms settling time (Figure 3-22). It was difficult to achieve less than 300 ms transient response without introducing instability to the system.

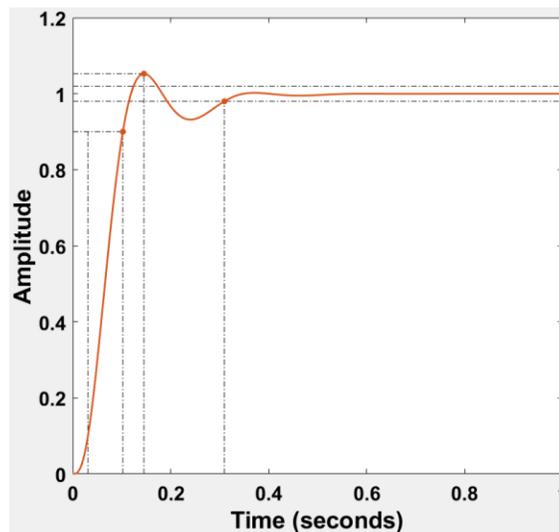


Figure 3-22. Step response PID controller

From the simulation, it was found that the integral part of the controller was essential for compensating for the in-rush current resulting from the starting of the blade motor. However, this effect was intended to be minimized in the field as the blade already running at full speed before the digging happens. Therefore, the PD controller was considered appropriate. If using a PD controller, the controller parameters become 164.9

and 5.74 for proportional and derivative parameter, respectively. The PD controller simulation has a comparable 360 ms settling time (Figure 3-23).

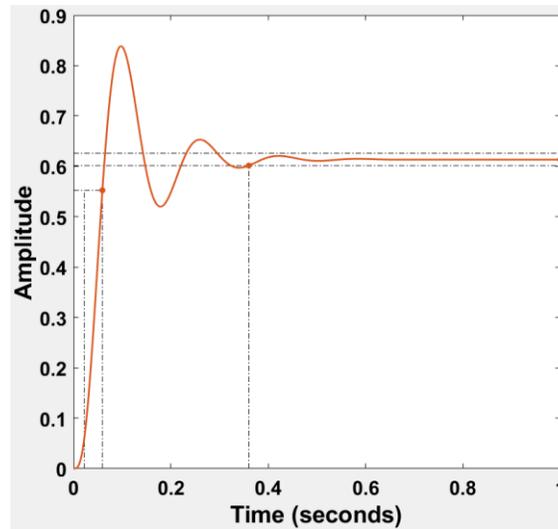


Figure 3-23. Step response PD controller

From the previous white-box controller simulation, the PD controller was considered suitable. In the black-box controller development, the desired settling time was improved to 200 ms with 20% overshoot. Root locus technique was used to design the PD controller from underdamped second-order system transfer function derived using black box modelling. The resulting controller parameters for the PD controller were: 4.7 and 0.07 for the proportional and derivative parameter, respectively. The comparison between the uncompensated and PD compensated system can be seen in Table 3-2 and graphically represented in Figure 3-24 and Figure 3-25, respectively. The result was the PD controller improved the steady state error considerably from 0.46 to 0.19 s.

Table 3-2. Comparison between uncompensated and PD compensated system

	Uncompensated	Simulation	PD Compensated	Simulation
Plant and controller	$\frac{K 327.7}{s^2 + 16.51s + 327.7}$		$\frac{K 327.7(0.07s + 4.7)}{s^2 + 16.51s + 327.7}$	
Gain (K)	0.02		0.07	
ζ	0.46		0.46	
ω_n	17.95		43.4	
%OS	20	20.5	20	25.5
T_s (s)	0.485	0.455	0.201	0.179
T_p (s)	0.193	0.190	0.082	0.065
Steady-state error	0.975		0.175	
Zero	none		-66.5	

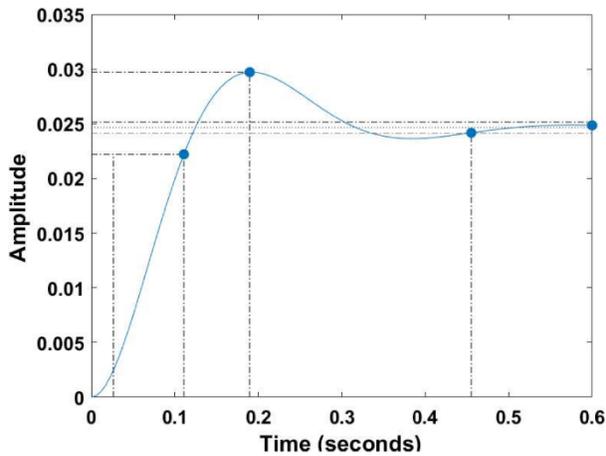


Figure 3-24. Uncompensated step response

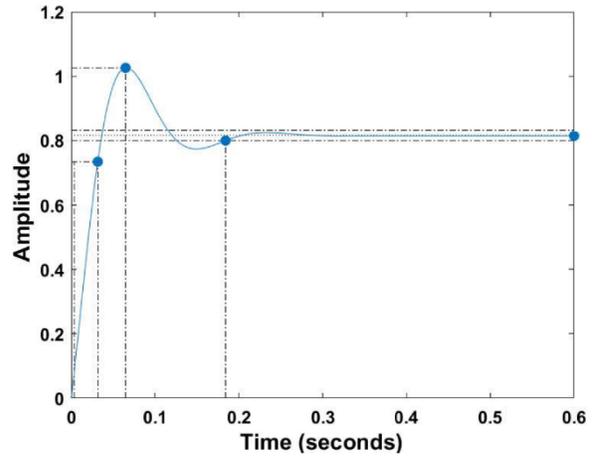


Figure 3-25. PD compensated step response

From the Ziegler-Nichols controller parameters (Table 3-1), PID controller parameter formula can be described as the following:

$$PWM \text{ rate} = K_p + \frac{K_p}{T_i} + K_p T_d \quad (3-14)$$

where the first, second and third summation elements are the proportional, integral and derivative controller parameter, respectively. Thus, the resulting PID controller parameters are 1.12, 3.73 and 0.084 for the proportional, integral and derivative controller parameter, respectively.

Controller Field Test

The controller parameters from both the white and black-box methods are summarized in Table 3-3. The results from both methods were tested to evaluate the control systems' performance. The 10 s digging test was held on a grass field (Figure 3-26) with relatively compacted soil (1123 to 1509 kPa cone index at 50 mm depth) to provide the worst-case scenario of the real-life sampling environment. The OSA controller modelling field test results are presented in Figure 3-27 to Figure 3-29. The controller formula used for this setup is the following:

$$PWM\ rate = 95 - \left(Pe(t) + I \int e(t)dt + D \frac{de}{dt} \right) \quad (3-15)$$

where 95 is the controller offset or PWM rate when the linear actuator has no movement, $e(t)$ is the error or the difference between the digging current setpoint and the blade digging current, while P , I and D are proportional, integral and derivatives controller parameters, respectively.

Table 3-3. Control design parameter summary

Control Design Method	Abbreviation	Controller Parameter		
		Proportional (P)	Integral (I)	Derivative (D)
White-Box Modelling	WB_PID	140.4	1473	3.104
	WB_PD	164.9	0	5.737
Black-Box Modelling Second-Order System	BB_PD	4.7	0	0.07
Black-Box Modelling Ziegler-Nichols	ZN_PID	1.12	3.73	0.084



Figure 3-26. OSA controller test at grass field

From Figure 3-27, the controller PWM output for both the white-box control design parameters (WB_PID and WB_PD) was too aggressive. This was mainly attributed to the use of the uncompacted bare sandy loam soil as the soil resistance model. Therefore, when both of the white box controllers were tested in the compacted soil with grass cover condition, the controller's PWM output overshoot and unstable. These controllers also exceeding both the threshold current set at 5 A and the breaker current at 50 A (Figure 3-28). This detrimental behavior affects the OSA physical structure as well the vehicle where the OSA is attached. The vibration was too violent and might cause premature wear or component failure. Figure 3-28 also indicates that the integral control parameter resulted in undesirable less hammering action while digging. In the same 10 s span, the WB_PID resulted in 3 digging current peaks which lasted approximately 1.5 s each, while the WB_PD resulted in 5 peaks for 0.7 s each. In the case of compacted soil, it is desirable to have more hammering action to break up clods and reduce the motor load. Therefore, from this test, a PD controller was determined to be desirable.

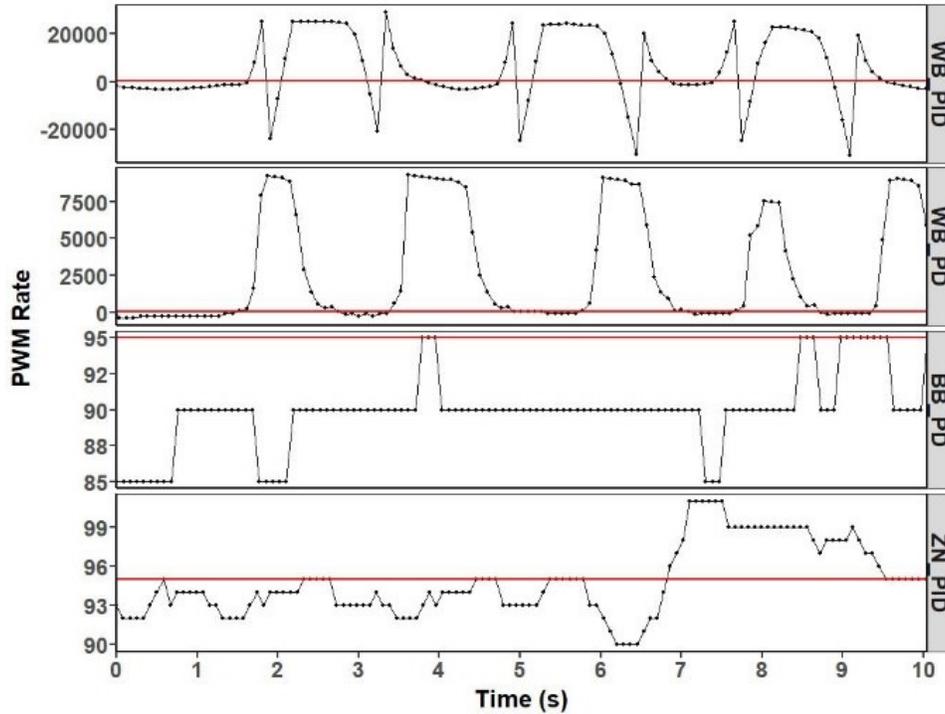


Figure 3-27. Controller output of the OSA control system during field validation test (WB : White-box, BB : Black-box, ZN : Ziegler-Nichols)

The black-box second-order control parameters (BB_PD) on the other hand, resulted in a maximum digging current of 5 A (Figure 3-28), matching the controller current setpoint. However, the digging time was significantly slower than the target of less than 15 s due to minimal PID controller error. This was caused by the grass cover which provided strong resistance and made the blade motor current increase by around 5 A before digging the soil. As a remedy, a larger controller setpoint was needed. Almost similar with the BB_PD, the Ziegler-Nichols controller output (ZN_PID) resulted in less than 5 A digging current (Figure 3-28). Interestingly, although it was still far from the target digging time, the Integral part of the controller was able to overcome the grass resistance and hence, able to provide enough hammering action to allow the soil digging process to occur. Uneven terrain conditions caused the OSA to dig at different starting depths (Figure 3-29).

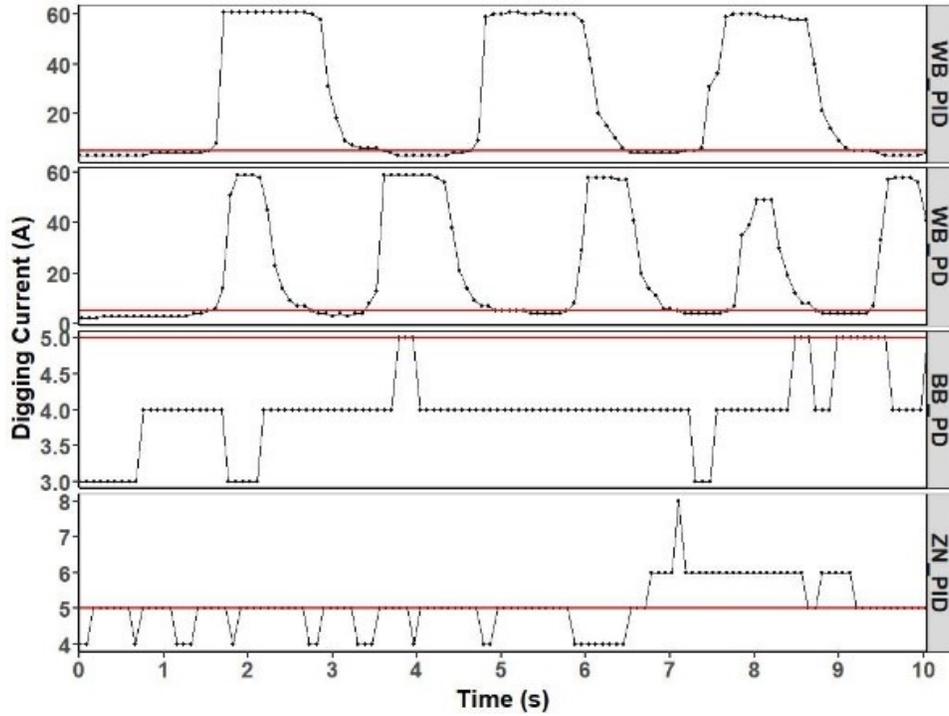


Figure 3-28. Digging current of the OSA control system during field validation test (WB : White-box, BB : Black-box, ZN : Ziegler-Nichols)

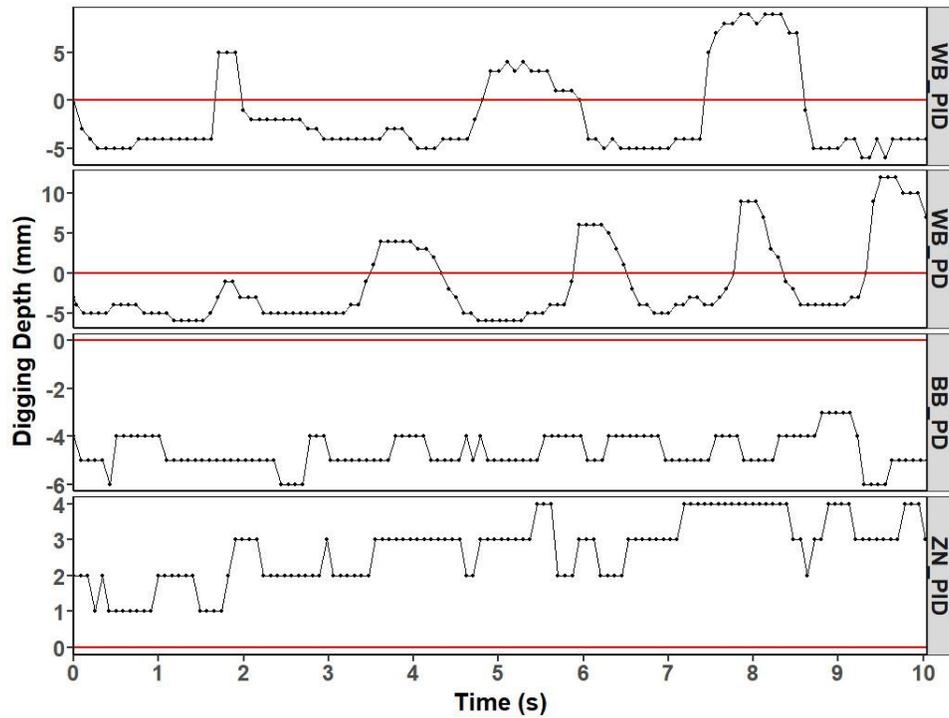


Figure 3-29. Digging depth of the OSA control system during field validation test (WB : White-box, BB : Black-box, ZN : Ziegler-Nichols)

Controller Field Tuning

Despite an intensive controller design effort, the digging system required more field tuning in a post-harvested corn field to represent typical field conditions and to achieve robust controller parameters. From the previous controller modelling field test, the PD controller was deemed appropriate for reliable OSA digging process as it resulted in repeated hammering actions and faster digging time. The test also provided valuable information that helps determine the initial controller tuning parameters.

While optimizing the controller parameters in the field, the appropriate values were found to be 9 and 2 for proportional and derivative parameter, respectively. The tuned controller output was in the range of frame linear actuator controller PWM signal (0 to 180) and able to provide repeated hammering actions when digging (Figure 3-30). The tuned parameters limit the digging current to less than 15 A (and Figure 3-31). The field tuned controller successfully achieved the design goal by digging 10 mm in 3 s and less than 15 s at 30 to 50 mm depths (Figure 3-32). The digging results example can be seen in Figure 3-33.

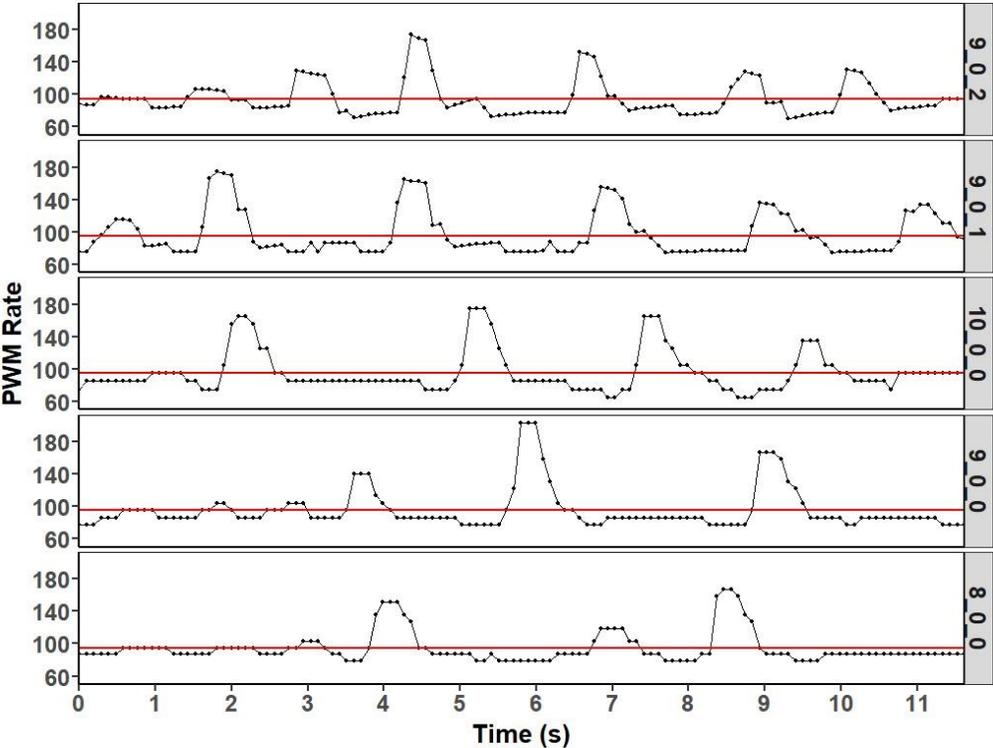


Figure 3-30. Controller output of the OSA control system during field tuning

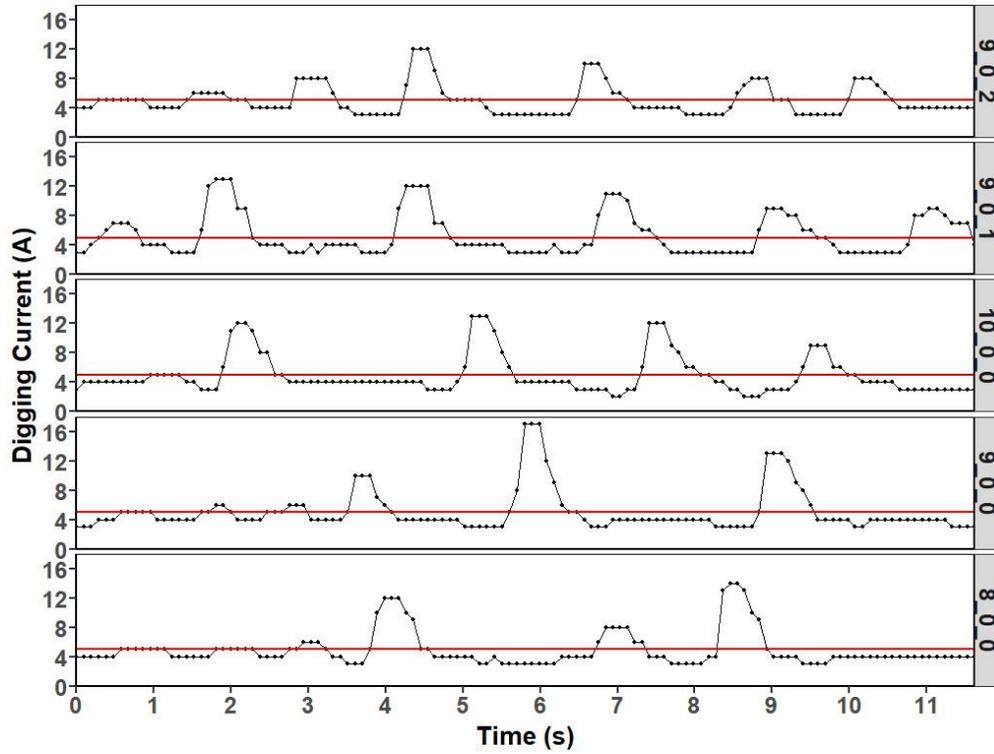


Figure 3-31. Digging current of the OSA control system during field tuning

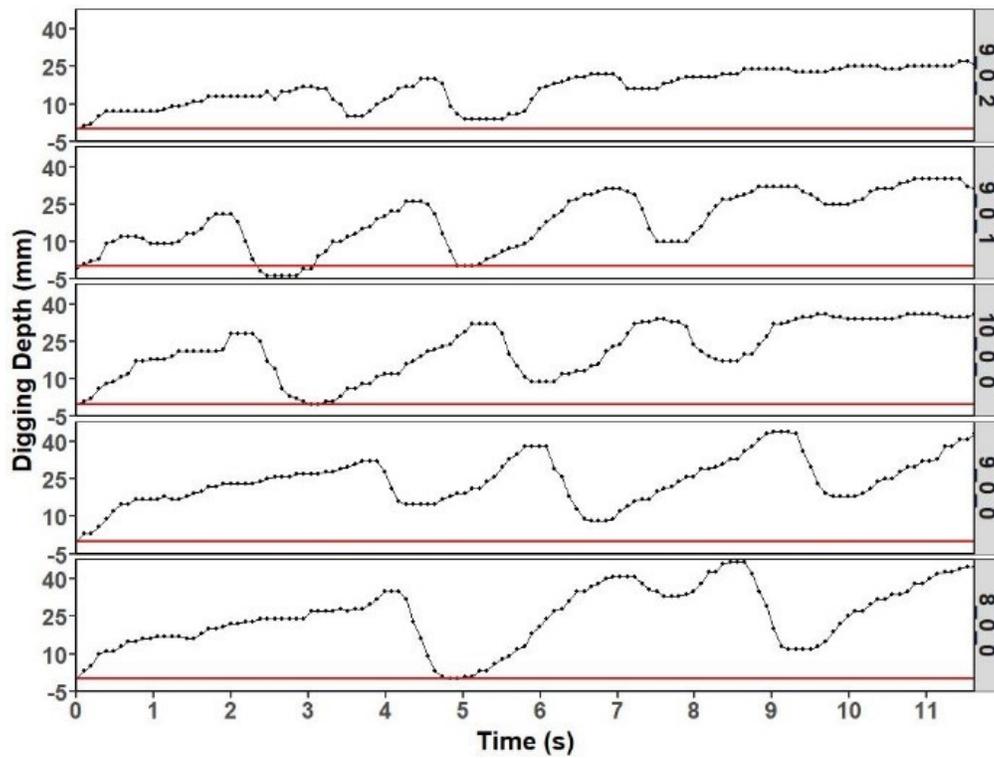


Figure 3-32. Digging depth of the OSA control system during field tuning



Figure 3-33. Typical field tuning digging result

3.2.6. OSA Field Operation

In the field setup, the OSA was attached to an all-terrain vehicle through a common drawbar towing hitch (Figure 3-34). The OSA successfully performed the entire sampling operation procedures. The digging result was able to provide an appropriate surface for the DSM using ISEs (Figure 3-35a and b). The hole covering action effectively restored the soil back into the sampling hole (Figure 3-35c). Finally, the cleaning process successfully removed sticking soil from the ISEs. Importantly, the OSA was equipped with a camera to monitor the digging blade position and soil conditions. This feature proved beneficial when the OSA was used in stony locations. The operator can override the operation anytime to protect the ISEs from breaking due to hitting the stones in soil.



Figure 3-34. The OSA field setup



(a) OSA digging result



(b) ISEs measurement



(c) OSA hole covering

Figure 3-35. The OSA field operation

3.2.7. OSA Field Test

The OSA control system was tested in the post-harvest soybean field number 22 at Macdonald Farm, Macdonald Campus, McGill University in November 2018. This field was selected to represent the typical agricultural field condition. The majority of the field consist of sandy loam with small area of loamy soil texture. The field compaction level ranging from 210 to 983 kPa cone index at 25 mm depth. The average volumetric water content was 21%. In addition, the field has undulating landscape with stony characteristics in some locations. Also, the post-harvest condition left a fair amount of soybean residue. All of these field characteristics was considered ideal to test the OSA control system performance.

The OSA direct soil measurement performance was tested to measure soil soluble nitrate and pH using ISEs at 16 experimental plots at the Emile A. Lods Agronomy Research Centre, Macdonald Campus, McGill University in July 2019. The plots were used to test the efficacy of biosolids amendment for corn production. The plots were treated with various level of nitrogen fertilizer (urea), compost, lime and household waste biosolids. Each of plots treatment was incorporated with the soil before planting and the OSA test was conducted between corn rows at V5 stage.

The OSA can be used with various ISE sensors suitable for DSM. In this field test, soil soluble nitrate and pH were selected because of the importance of nitrogen fertilization and liming in agriculture. Antimony (Veris Technologies Inc., Salina, Kansas, USA) combination ISE was used for pH measurements. Soluble nitrate was measured using

nitrate combination ISE (Nico Sensors, Huntingdon Valley, PA, USA). Before the field measurements, each ISE was calibrated. pH ISE was calibrated using pH 4, 7 and 10 standard calibration solution while 10, 100 and 1000 ppm nitrate calibration standard were used to calibrate the nitrate ISE.

Each ISE's measurement was repeated two times within 1 m from the first sampling location. At each sampling location, a scoop of soil was gathered to be further analyzed in the laboratory. For nitrate analysis, the moist soil sub-samples (5 g) were mixed with 40 ml 2 mol/L KCl, shaken for 30 minutes and filtered using Fisherbrand Q5 filter paper (Thermo Fisher Scientific Inc., Waltham, Massachusetts, USA). This nitrate extraction process was started as soon as the OSA field test finished. The filtrated soil suspension then analyzed colorimetrically using Lachat QuikChem® 8500 FIA+ flow injection autoanalyzer (Hach LLC, Milwaukee, Wisconsin, USA) for soil soluble nitrate content (Mulvaney, 1996). The soil moisture content was determined using standard oven-dry method. For the soil pH analysis, the air dried soil sub-samples were sieved using 2 mm sieve, diluted using deionized water until 1:1 soil/extractant ratio was achieved, shaken for 30 minutes and allow to settle for 15 minutes. The clear supernatant then measured using a glass pH combination ISE (Thomas, 1996). Small discrepancy in laboratory soil nitrate determination and DSM was expected because the ISE measures nitrate ionic activity from soil water directly, whereas in the laboratory, the KCl extraction replaces the exchangeable nitrate ions that adhere on the soil material. However, similar measurement trend is expected (Mulvaney, 1996). The ISEs measurement precision (repeatability) was assessed using root mean squared error (RMSE) method, whereas the accuracy error was evaluated as standard error of predicted measurement compared with the standard laboratory analysis result (Sethuramasamyraja et al., 2008).

3.3. Results and Discussion

3.3.1. OSA Controller Performance

The OSA control system testing started by digging several holes to determine the effective digging depth in the field. The OSA GUI ease the user to perform specific task (e.g. digging only). A 15 mm digging depth was considered sufficient to provide a proper soil surface condition for DSM using ISEs. One example of OSA digging control output can be seen in Figure 3-36. In this particular location, the OSA completed the digging process in 3.7 s and thus, the overall OSA operation took less than 40 s. This result was attributed to the ideal field conditions identical to what was found in Figure 3-33. The soil was not stony and had a considerable compaction of 702 kPa cone index at 25 mm depth.

The digging control system successfully limited the digging current far below the breaker current of 50 A. It was also noticed that the frequent hammering action was less apparent. The digging controller was able to regulate the movement of the frame linear actuator according to the lower soil compaction level. Thus, this reduced the digging time by limiting the return movement of the frame after hitting the soil.

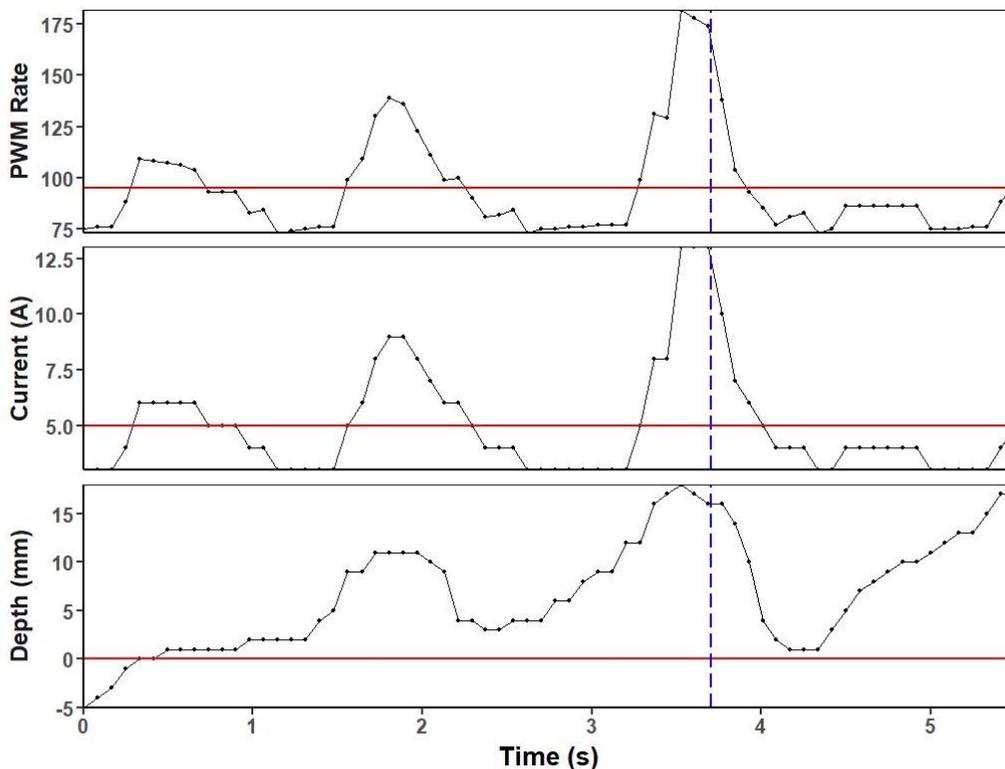


Figure 3-36. OSA digging control output in ideal field condition

In stony and high soil compaction locations (948 kPa cone index at 2.5 cm depth), it took 20.8 s to complete the digging process (Figure 3-37). In this situation, there were frequent current spikes observed due to the impact with compacted soil structures or small stones. As a result, the digging action was oscillating. This movement was aggravated by the slow blade motor current sensor acquisition speed of 84 ms with a median filter of 5 consecutive measurements to provide reliable data. The reason for filtering the current data was to provide a smooth reading for continuous digging. On the other hand, it reduced the OSA control response in this type of soil.

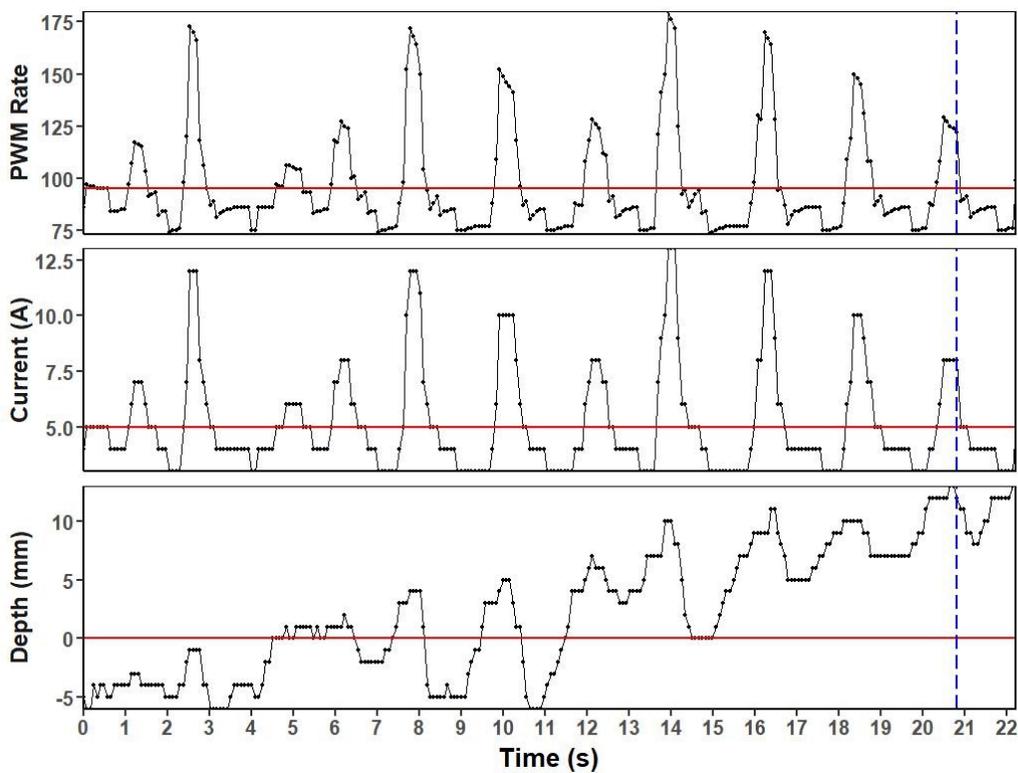


Figure 3-37. OSA digging control output in stony locations

In stony locations, the digging controller was able to regulate the frame linear actuator to create more frequent impact movements, which was desired as it results in the hammering action to break up clods and remove stones. Therefore, the optimized digging controller reduces the blade motor load by performing multiple impacts to the soil. Additionally, although the OSA was tested in less ideal stony conditions, the control system was able to limit the maximum digging load to less than 50 A. The control system

also provided a fast soil preparation time. Ultimately, the total OSA operation time required around 60 s to complete.

3.3.2. OSA Direct Soil Measurement Assessment

The OSA data acquisition system was able to provide reliable georeferenced soil chemical data without electrical noise disturbance or cross-referencing among the ISEs. The RMSE for pH and NO₃ ISEs were 0.43 pH and 0.16 pNO₃ (12.4 ppm), with standard error of measurement of 0.55 pH and 0.25 pNO₃ (10.9 ppm), respectively. The antimony pH and NO₃ ISEs had a decent calibration performance and satisfactorily characterizing the soil pH and nitrate with R² of 0.59 (Figure 3-38) and 0.72 (Figure 3-39), respectively. The soil pH and nitrate prediction were diverse ranging from 4.8 to 9.0 pH and from 2 to 44 ppm NO₃, respectively. The nitrate prediction was in accordance with the typical range of soil nitrate found during growing season of 0 to 140 ppm (Jahn et al., 2006).

The RMSE and standard error of the pH ISE was acceptable compared to what found by Adamchuk et al. (2005) for a typical DSM. They used glass membrane pH ISEs and found that the RMSE was from 0.19 to 0.21 pH with 0.11 to 0.12 pH standard error. The inferior performance was attributed to the difference in the pH ISE's sensitive membrane type. Although antimony pH ISE was more robust compared to a glass membrane, the antimony ISE has more uncertainty related to surface oxidation as found by Conkling and Blanchar (1988) and Michael et al. (2011). Better performance coming from nitrate ISE with comparable accuracy to result found by Sethuramasamyraja et al. (2007) at 2:1 soil to water ratio of 0.23 pNO₃. Nitrate ISE also had a better RMSE compared to agitated soil measurement (ASM) result of 0.22 pNO₃ (Sethuramasamyraja et al., 2008) and on-the-go DSM result of 0.43 pNO₃ (Adamchuk et al., 2005).

From the laboratory analysis, the average field gravimetric soil water content was 15.2%. Although the field water content was in the range for typical DSM of 15 to 30% (Adamchuk et al., 1999), still it was considered less than ideal. The benefit of the new OSA GUI was that it allowed the user to manually perform ISEs washing. Therefore, the user can utilize the washing action to wet the ISEs before deploying it to compensate for dry soil conditions. Although during the field test, the user already performed ISEs

washing before the measurement, since the soil sample was collected in bulk, the laboratory gravimetric soil water content was less when compared to the ISEs membrane area soil water content. Therefore, the ISEs calibration line was shifted (Zhang and Wienhold, 2002). Furthermore, this effect was amplified for the soil nitrate estimation as no chemical ion extraction took place in the field, which is the opposite of the laboratory protocol.

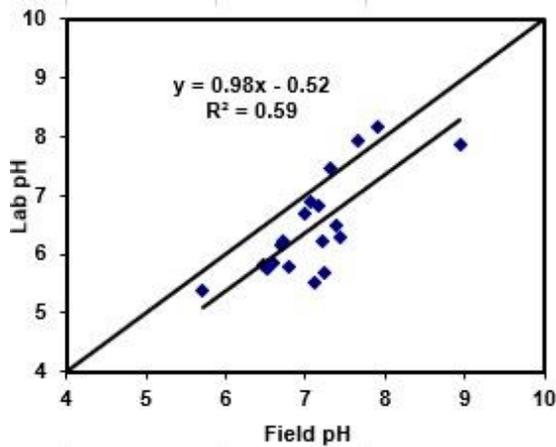


Figure 3-38. OSA pH ISE field calibration

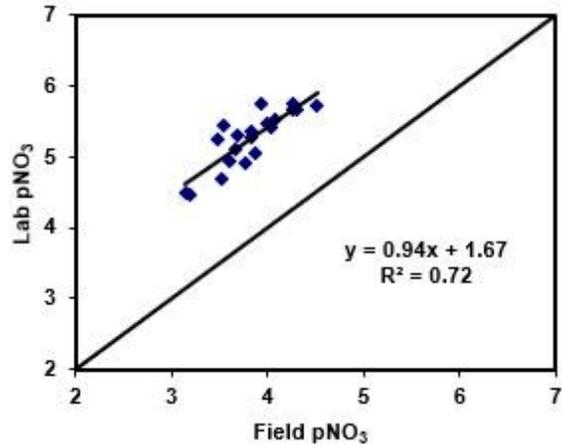


Figure 3-39. OSA nitrate ISE field calibration

From the test result, the optimized digging process produce suitable condition for various soil chemical measurement using ISEs. As the OSA DAQ support up to four ISEs, this allow for soil macro-nutrients (NPK) and soil pH estimation simultaneously. DSM of soil pH using ISE is based on inherent soil moisture and will provide information about soil acidity and the necessity of liming, not the quantity of lime needed (Adamchuk et al., 2004; Adamchuk et al., 1999). Therefore, DSM using pH ISE cannot be used solely to determine soil liming requirement. Instead, a soil buffer pH measurement is needed. Nevertheless, DSM of soil pH still can be used together with other sensors such as EC_a sensor to determine soil properties associated with soil buffering capacity (CEC, texture, organic matter content) (Doolittle and Brevik, 2014; Adamchuk et al., 2007). Thus, these sensors combination will improve the liming recommendation (Lund et al., 2005; Michael et al., 2011; Wienhold and Doran, 2008).

The OSA was designed to be a versatile soil sensing platform which can be equipped with various portable soil sensors not only ISE, for example: handheld microscope

(Sudarsan et al., 2018) and spectroscopy sensors (Angelopoulou et al., 2020). These sensors will provide additional information to improve the soil macro-nutrients estimation from DSM using ISEs (La et al., 2016; Sinfield et al., 2010). Soil sensors data fusion will improve other soil properties prediction as well as farm management practices (Mahmood et al., 2012).

The on-the-spot measurement characteristic of the OSA required the platform to stop while conducting soil measurement. This methodology does not require the OSA to sample in transect arrangement like other on-the-go platforms. Furthermore, with comparable soil measurement time, the OSA is expected to deliver similar density georeferenced data with less time. Other benefit is that the OSA was designed to be vehicle mounted soil sensing platform. This feature allows the OSA to be mounted on various vehicles including a robot. Thus, creating a fully autonomous or remotely controlled soil sensing platform. The OSA sampling locations can be arranged using the directed sampling method from the inverted EMI sensor data explained in Chapter 5.

3.4. Conclusion

The new OSA design was successfully conducted direct soil measurements of multiple soil chemical properties. The optimized digging controller was able to manage the digging load under various field conditions. It provided an efficient digging operation and better digging load management. Typically, for each sampling location, the complete OSA operation took 60 s to complete. Both the antimony pH and nitrate ISEs were performing satisfactorily when predicting soil pH and nitrate with R^2 of 0.59 and 0.72, respectively, with RMSE of 0.43 pH and 0.16 pNO₃. Ultimately, the new OSA was able to provide a robust platform for multiple soil chemical property measurements.

Connection to Chapter 4

After developing the OSA, a portable alternative soil sensing platform needs to be developed to accommodate farmers with small and scattered fields. The portable soil sensing platform should support various ISE sensors for DSM and have a soil sampler mechanism for laboratory analysis. To successfully apply VRA, multiple ISE sensors are necessary to quantify major soil properties, such as soil pH and nitrate. Previously, the manual sampler was developed to hold a single ISE sensor (Adamchuk, 2005). This time, the data acquisition system was based on a Whitebox Labs' Tentacle (Meister Whiteboxes GmbH, Basel, Switzerland) with four Atlas Scientific ORP (oxidation-reduction potential) circuit (Atlas Scientific LLC, Brooklyn, New York, USA) which is able to support up to four ISE sensors compared to the previous FieldScout pH 110 Meter (Spectrum Technologies, Inc., Aurora, Illinois, USA) which only supports one ISE. The soil sampler still used a modified generic soil sampler (JMC, Clements Associates Inc., Newton, Indiana, USA) which has been proven to be reliable. The research has been published in AESAP conference 2016 in Indonesia and ISPA conference 2018 in Montreal, Canada.

Leksono, E. and V.I. Adamchuk. 2016. Development of a portable multiple ion-selective electrodes apparatus for rapid soil nitrate measurement. In: Proceedings of the 1st International Conference on the Role of Agricultural Engineering of Sustainable Agriculture Production (AESAP), Bogor, Indonesia, 13-14 December 2016, 153-164. Bogor, Indonesia: Bogor Agricultural University.

Leksono, E., V. Adamchuk, J. Whalen, and R. Buelvas. 2018. Development of a manual soil sensing system for measuring multiple chemical soil properties in the field. In: Proceedings of the 14th International Conference on Precision Agriculture, Montreal, Quebec, Canada. 24-27 June 2018. International Society of Precision Agriculture (published on-line at <http://www.ispag.org>, 5 pages).

4. Development of Manual Soil Analyzer

Abstract

Variable Rate Fertilizer Application (VRA) requires soil chemical data. One of the preferred methods for analyzing soil chemical properties in the field is by using ion selective electrodes (ISEs). To accommodate portability in soil measurements, a manual soil sensing system was developed. Nitrate and pH ISEs were integrated in order to provide a general outlook on the condition of essential soil nutrients. These ISEs were placed on a modified hand-held soil sampler equipped with variable depth adjustment, water reservoir, hand pump and spray nozzles for rinsing the ISEs. An Arduino shield from Whitebox Labs' Tentacle (Meister Whiteboxes GmbH, Basel, Switzerland) with four Atlas Scientific ORP (oxidation-reduction potential) circuit (Atlas Scientific LLC, Brooklyn, New York, USA) was used as the data acquisition (DAQ) system. A Global Positioning System (GPS) data logger and a temperature sensor were added to provide more input for the VRA development. A Bluetooth module was added to the DAQ to enable real-time ISEs response monitoring through a generic mobile phone Bluetooth terminal application. Additionally, the DAQ was designed to have one button operation to provide a simple user interface. The DAQ system was successfully tested for noise, cross talk and ground loop errors. In two field tests, both ISEs performed satisfactorily with R^2 of 0.53 to 0.88 and RMSE ranging from 0.17 to 0.36 pH when predicting soil pH and R^2 of 0.83 to 0.84 with RMSE of 0.21 to 0.29 pNO_3 for soil nitrate prediction. Overall, the new manual soil sensing system would provide a portable alternative for depicting multiple soil chemical variability in smaller fields.

Keywords. manual soil sampler, direct soil measurement (DSM), ion-selective electrode (ISE), nitrate, pH.

4.1. Introduction

Information about soil nitrate and pH status is important to provide an outlook on the condition of soil nutrients in the field. Furthermore, by knowing that information, farmers can adjust their farming regime to balance soil nutrients for crop needs, reducing the possibility of severe environmental problems and inefficient expenditure from over fertilization (Liu et al., 2015). This soil nutrient information is becoming indispensable, especially for small scale farmers who usually have small and scattered fields. Moreover, they account for the majority of farmers in developing countries (Jelsma et al., 2019; Kubitzka et al., 2018) where they often practice unsustainable nomadic style farming and rarely intensify their field (Vosti et al., 2001).

Variable rate fertilizer application (VRA) is one way in which precision agriculture offers field intensification. Critical soil chemical information, such as soil pH and nitrate together with their georeferenced sampling locations, is essential for VRA input parameters (Srinivasan, 2006; Fleming et al., 2000; Baxter et al., 2003). These inputs can be processed, analyzed, and modelled to create fertilizer recommendation maps (Van Alphen and Stoorvogel, 2000; Dillingham et al., 2012). Information about soil pH and nitrate can be quantified using ion-selective electrodes (ISE). These sensors have been successfully used for analyzing soil pH and nitrate either in the laboratory (Mulvaney, 1996; Thomas, 1996) or directly in the field (Adamchuk et al., 1999; Adsett et al., 1999). Other advantages of ISEs are that they are small, portable, and only need deionized water as a solvent.

Various efforts have been employed in determining fertilizer prescription maps. One of the methods popularly used is an on-the-go soil method using direct soil measurement (DSM) (Adamchuk et al., 2004). However, this method requires a vehicle to operate, which may be difficult to be implemented by small scale farmers with small, scattered fields. Portability becomes a major design factor to fulfill the need of small scale farmers and to accomplish the goal as an initial soil measurement tool. Previously, a portable manual soil sampler was developed to hold a single ISE sensor (Adamchuk, 2005). This manual platform was based on a modified generic soil sampler (JMC, Clements Associates Inc., Newton, Indiana, USA). Furthermore, the Data Acquisition System (DAQ)

was based on a FieldScout pH 110 Meter (Spectrum Technologies, Inc., Aurora, Illinois, USA) which was quite expensive and only supports one ISE sensor. Similar to manual soil sampling, the operator pushes the platform into the soil to 15 to 30 cm depth to obtain a representative soil sample (Sudduth et al., 2013). Then, the ISE was brought into contact with the extracted soil to measure the soil chemical concentration of interest. The operator can monitor the ISE reading from the FieldScout pH 110 Meter display. Finally, manual pumping action sprays water towards the ISE to provide a cleaning action.

An affordable and multichannel DAQ needs to be developed to provide an economic benefit for farmers as well as supporting multiple ISEs measurements. Various efforts have been conducted to fulfil those requirements. For example Wang et al. (2011) used an 8 channel ISE DAQ system consisting of a CMOS analog multiplexer switch DG407, high impedance-low input bias amplifier INA116, 16 bit ADC AD7705, and ultra-low power MSP430F1611 MCU. A voltage follower circuit may also be added to further improve the amplifier impedance (Jianhan et al., 2007). Nowadays, the popular and affordable Arduino Uno microcontroller, which is based on the ATmega328P chip, provides an opportunity to develop a simple and reliable multichannel DAQ. Moreover, there is less information on the portable manual soil sampler and analyzer for multiple soil chemical properties measurement.

The objective of this research was to develop and test a manual soil analyzer for measuring multiple soil chemical properties in the field using ISEs. This probe should be suitable for rapid quantification of spatial soil variability and to guide site-specific management of selected agricultural inputs in small plots and specialty cropping system scenarios.

4.2. Materials and Methods

4.2.1. Development of Soil Sampler

SolidWorks 2015 (Dassault Systemes S.A., Waltham, Massachusetts, USA) was used to create the conceptual design of the manual soil sensing platform (Figure 4-1). A generic 90 cm long soil sampler (JMC, Clements Associates Inc., Newton, Indiana, USA) was used as a base for developing the soil sensing platform. The 3D printed spring-loaded ISE clamp was designed to hold four ISE bodies (assuming 12 mm diameter of each probe) for multiple soil chemical measurements. Four TeeJet XR11002VK nozzles (TeeJet Technologies, Glendale Heights, Illinois, USA), water bottle (approximately 1500 ml) and manual piston hand pump were used to clean the ISEs. Water nozzles are angled 8° towards the ISE to provide effective ISE cleaning.

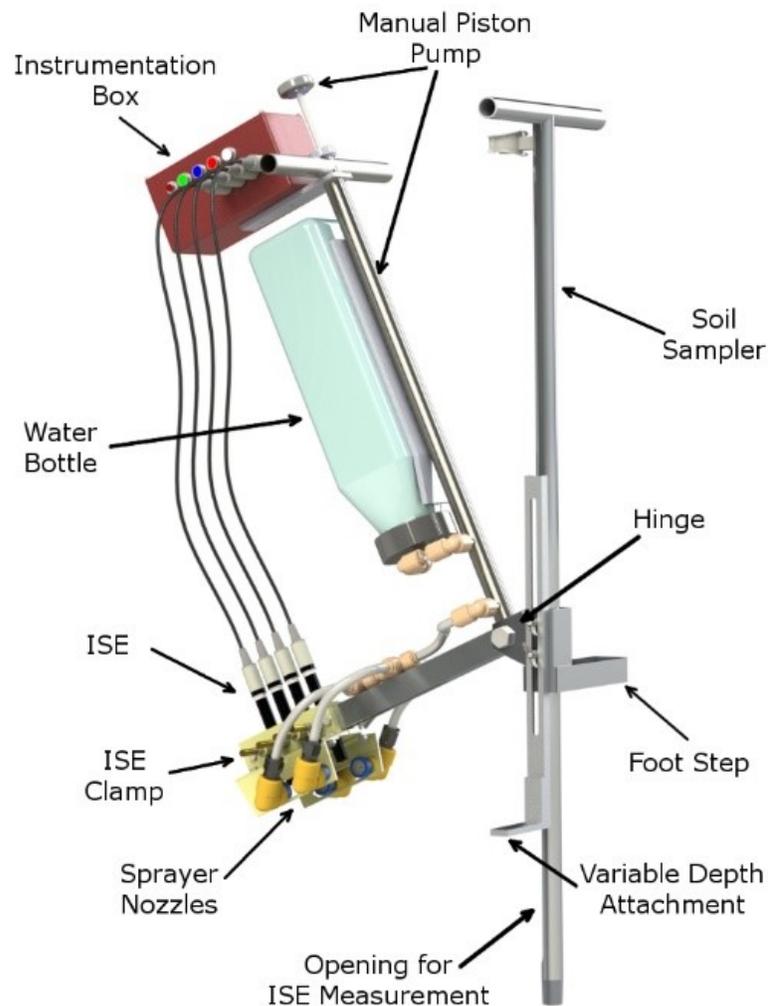


Figure 4-1. Conceptual design of the apparatus

4.2.2. Electronics and Data Acquisition System (DAQ)

A four channel BNC circuit Arduino shield from Whitebox Labs' Tentacle (Meister Whiteboxes GmbH, Basel, Switzerland) was used as a base for the data acquisition system. A DS18B20 soil temperature sensor, a GPS receiver and data logger (Adafruit Industries, Manhattan, New York, USA) were added to the DAQ to provide more input for the VRA development. A soil temperature sensor feature is valuable to provide indirect estimation of soil water content (Lakshmi et al., 2003; Hanks, 1992), which is needed as was determined from experience from the DAQ developed in Chapter 3. Additionally, a generic Bluetooth transceiver module was incorporated into the DAQ to enable real-time ISEs response monitoring through a generic mobile phone Bluetooth terminal application.

The DAQ was designed to have instrument nulling to remove systematic error from stray voltage when not measuring (Wheeler and Ganji, 2010). This functionality and starting the ISEs measurement are incorporated into the one button operation to provide a simple user interface. The user needs to depress and hold the start button for 5 s to access the nulling functionality and a short press on the start button to start the GPS reading and subsequently, conduct the ISEs measurement. The workflow of the DAQ system can be seen in Figure 4-2.

To measure multiple soil chemical properties using ISEs, four Atlas Scientific ORP (oxidation-reduction potential) circuits (Atlas Scientific LLC, Brooklyn, New York, USA) were installed. Two models of ORP circuits were tested: customized 100 ms and standard 1000 ms sampling rate (further called the first and second Tentacle, respectively). For the first Tentacle, 100 ms was determined to be the optimal sampling time for each ORP circuit while for the second Tentacle, a query time of 1000 ms for each circuit proved to be sufficient to provide a stable data stream. Previously, the one ISE FieldScout pH 110 Meter does not have a time series ISE response logger. The newly developed DAQ with ORP circuits has a faster sampling rate and continuous time series output which is important to evaluate ISEs response while measuring. Both Tentacle DAQ were tested for noise and cross talk isolation. Additionally, ground loop test was conducted to assess their applicability for multiple DSM using mono ISEs.

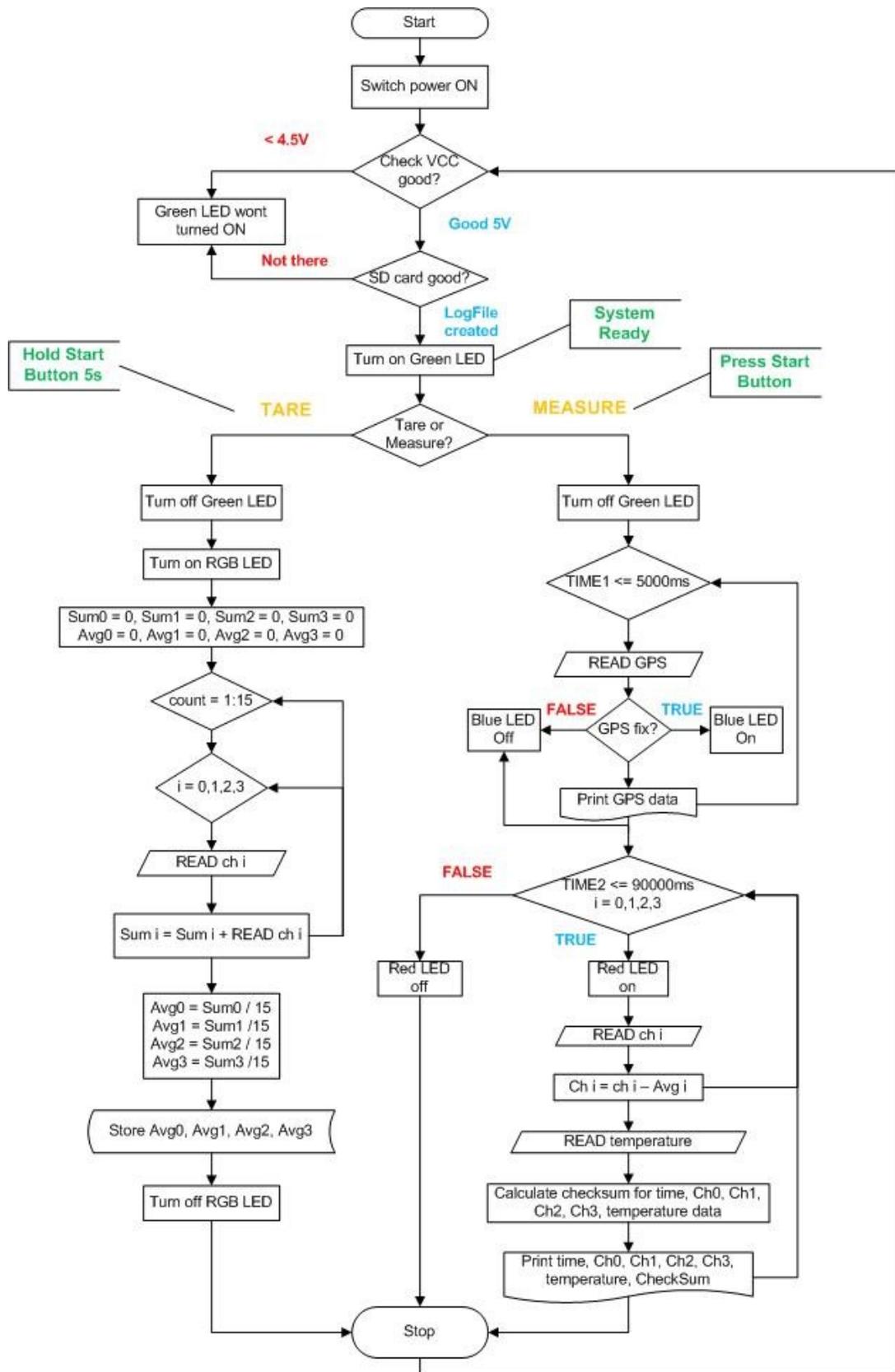


Figure 4-2. Manual soil chemical analyzer and sampler DAQ flow

DAQ Noise Test

Noise in the digital circuit is described as signal deviation from its nominal value (Vasilescu, 2006). For noise tests, the Tentacle shield was connected through the USB cable to a laptop and alternately plugged into an AC power line to the laptop. Then the ISE voltage signals were recorded and observed for any deviations. In this test, a glass pH combination ISE was used with pH 4 calibration solution.

DAQ Crosstalk and Ground Loop Test

Both Tentacles were tested for crosstalk to assess the measurement error caused by interference from adjacent channels (Wheeler and Ganji, 2004). A ground loop error test was conducted by connecting two or more circuits (which have their own designated ground) together (Vasilescu, 2006). The crosstalk and ground loop error testing were done by alternately dipping each ISE, or combination ISEs, into its calibration solution for 90 s (Table 4-1). Antimony and glass pH combination ISEs were plugged to Tentacle channel 1 and 3, respectively, while nitrate combination and mono ISEs were plugged to channel 2 and 4, respectively (Table 4-2). The calibration solutions used in this test were buffer pH 4 and 1000 ppm NaNO₃. Before the test, all ISEs were conditioned by dipping them into their calibration solutions for 1 h. The resulting time series data from the testing were plotted together to assess the effect of channel cross talk.

Table 4-1. Tentacle channel 1 crosstalk test sequence

Event number	ISE plugged to channel	Channel plugged with BNC cover
1	1	2, 3, 4
2	1, 2	3, 4
3	1, 3	2, 4
4	1, 4	2, 3
5	1, 2, 3	4
6	1, 2, 4	3
7	1, 3, 4	2
8	1, 2, 3, 4	-

Table 4-2. ISEs used in the crosstalk test

ISE id	Tentacle channel (ch)	ISE	Reference
pH_Sb_F	1	Sb, combination	integrated
N_F	2	PVC, combination	integrated
pH_G_F	3	Glass, combination	integrated
N_H	4	PVC, mono	N_F

4.2.3. Field Operation

The operation of the apparatus started by conducting instrument nulling/tare by covering all four DAQ channels with a BNC plug and proceeded by pressing the start button for 5 s to enter the tare procedure. The DAQ makes an average of 15 sets of ISEs measurements from each channel. The resulting tare value will be used to subtract the ISEs measurement when sampling. After conducting the instrument nulling, ISEs can be plugged into the dedicated DAQ channels and the apparatus is ready for soil sampling.

Soil sampling procedures are started by pressing the soil sampler into the soil at the pre-set depth (Figure 4-3a). Then, the hand pump handle is operated so that the ISEs have firm contact with the soil sample (Figure 4-3b). The soil measurement can be started immediately by a short press on the DAQ start button. Then, the DAQ will start collecting GPS data for 5 s and four ISEs measurements for 90 s. After each soil measurement, the ISEs are cleaned using a manual pump-sprayer (Figure 4-3c). The sampled soil was removed automatically at the next sampling location during the insertion of the probe into the soil.



Figure 4-3. Multiple ISEs manual probe operation.
(a) soil sampling, (b) soil measurement, (c) ISE cleaning

4.2.4. Field Test

The manual soil sampler was tested to measure soil soluble nitrate, phosphate and pH using ion-selective electrodes (ISEs) at 12 and 10 locations in organic amendment and 4R (Right Source, Right Rate, Right Time, Right Place) Nutrient Stewardship Nitrogen fertilizer experimental plots, respectively. The first experiment was conducted in September 2018 while the second was in July 2019, both at the Emile A. Lods Agronomy Research Centre, Macdonald Campus, McGill University. In the first experiment, glass (Cole-Parmer Instrument Company LLC, Vernon Hills, Illinois, USA) and antimony (Veris Technologies, Inc., Salina, Kansas, USA) combination ISEs were used to measure soil pH, while soluble nitrate was measured using a mono PVC Y360 (360 Yield Center, Morton, Illinois, USA) ISE.

In the second experiment, a scalloped epoxy pH (Cole-Parmer, Vernon Hills, IL, USA) combination ISE was used to measure soil pH. Soluble nitrate was measured using a combination nitrate ISE (Nico Sensors, Huntingdon Valley, PA, USA). Before field measurements, each ISE was calibrated. The pH ISEs were calibrated using pH 4, 7 and 10 calibration solutions, while the nitrate ISE was calibrated using 10, 100 and 1000 ppm N-nitrate. Each measurement was repeated two times under 1 m between sampling locations. To represent reference measurements, at each sampling location, the soil from the manual soil sampler was collected to be further extracted using 2 mol/L KCl for soil nitrate and analyzed using Lachat QuikChem® 8500 FIA+ (Lachat Instruments, Milwaukee, Wisconsin, USA) for soil soluble nitrate content. Soil pH was analyzed in a 1:1 soil to water dilution using a glass pH combination ISE (Sparks et al., 1996). The laboratory soil extraction and measurement started directly after the field sampling. Uncertain measurements due to questionable contact with soil and/or the potential for inefficient electrode cleaning were removed from the dataset. The ISEs measurement precision (repeatability) was assessed using root mean squared error (RMSE) method, whereas the accuracy error was evaluated as standard error of predicted measurement compared with the standard laboratory analysis result (Sethuramasamyraja et al., 2008).

4.3. Results and Discussion

4.3.1. Tentacle Noise

The noise test results for both Tentacles are displayed in Figure 4-4 and Figure 4-5. From Figure 4-4, the first Tentacle reading fluctuated as much as ± 40 mV while the laptop charger was plugged in. Fluctuation by as much as ± 3 mV from the reading mean was also observed while the laptop was using the battery. This also shows that the first Tentacle has a high noise floor. The suspected noise source was from our laboratory AC line since a lot of high wattage appliances were in operation at that time. The noise affected the USB port where the Arduino was plugged in and altered the nominal reading. It was shown in Figure 4-4 that the first Tentacle cannot filter this noise. On the contrary, from Figure 4-5, there was no reading deviation observed in the second Tentacle when the laptop was using a battery or it was plugged in. This also indicates that the second Tentacle has a low noise floor suitable for both laboratory and *in-situ* ISEs measurements. The drifting ISE reading observed from Figure 4-5 was attributed to an old ISE and coaxial cables and it was not attributed to the DAQ. Therefore, it was clear that the second Tentacle provides better electrical noise protection.

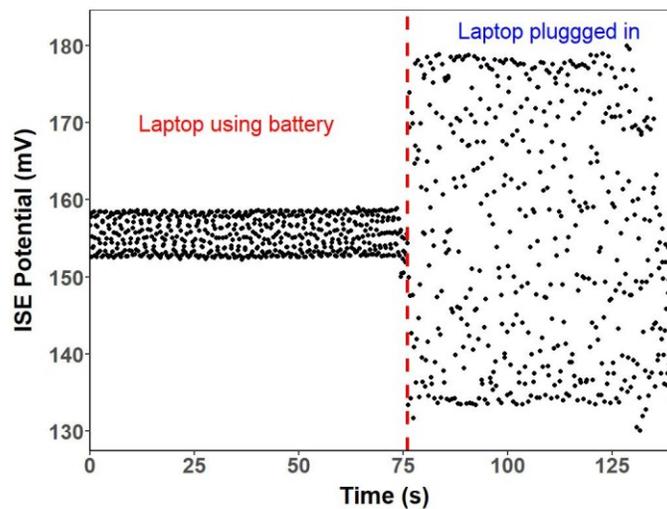


Figure 4-4. Tentacle noise test for the first Tentacle (100 ms sampling rate)

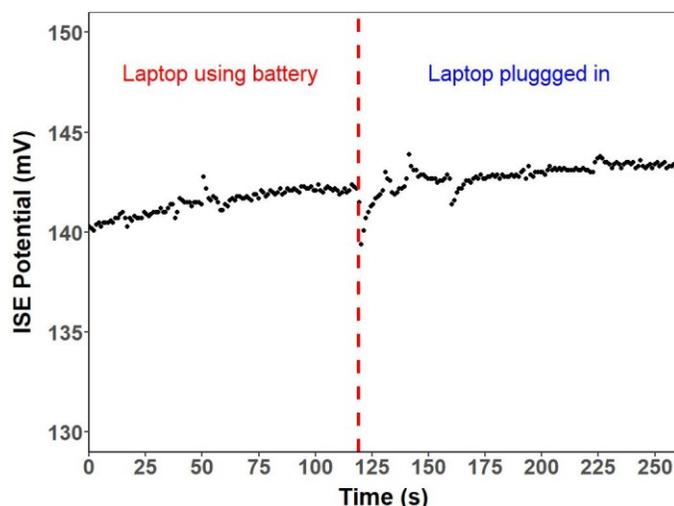


Figure 4-5. Tentacle noise test for the second Tentacle (1000 ms sampling rate)

During the nitrate mono ISE calibration using the first Tentacle (shared reference with glass combination pH ISE), the pH ISE reading fluctuated as a result of operator movement (Figure 4-6). The ISE reading spiked up to 8 mV when the operator moved closer to the ISEs and the first Tentacle was then followed by stabilization when operator stopped moving. A steep decrease by as much as 9 mV happened when the operator moved away from the system. Interestingly, there was no reading fluctuation from operator movement on the antimony pH combination or nitrate combination ISEs signal. This finding shows that the first Tentacle experienced ground loop and does not have proper channels isolation. Therefore, it is not suitable for ISEs with a shared reference.

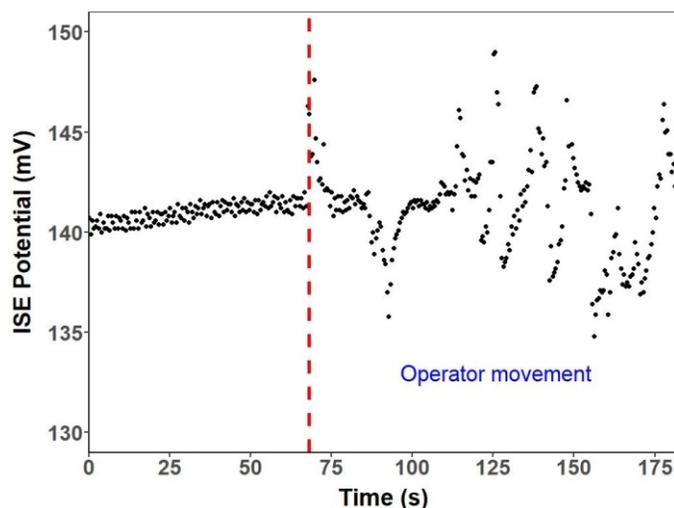


Figure 4-6. Effect of operator movement on the first Tentacle reading when using shared reference ISE

4.3.2. Tentacle Cross Talk and Ground Loop

Figure 4-7 shows the first Tentacle cross talk result. Channel 1 did not experience any cross talk from all adjacent channels. The sudden increase in ISE potential was caused when plugging/unplugging the ISE and it was normal. Interestingly, at event 6 (700 to 800 s), when channels 1, 2, and 3 were dipped into their calibration solutions, the measured potential for channel 2 was -11.9 mV and then rapidly changed to 19.2 mV when measuring channels 1, 2 and 4. The suspected cause was the ground loop from the shared reference between full-cell combination and half-cell mono nitrate ISE. This finding supports our previous observation that the first Tentacle was not properly isolated and not suitable for ISEs with a shared reference. Therefore, this limits our ISEs selection to a full-cell combination ISE only. This is not beneficial economically because the half-cell mono ISE is less expensive than the full-cell combination ISE (Sethuramasamyraja et al., 2008).

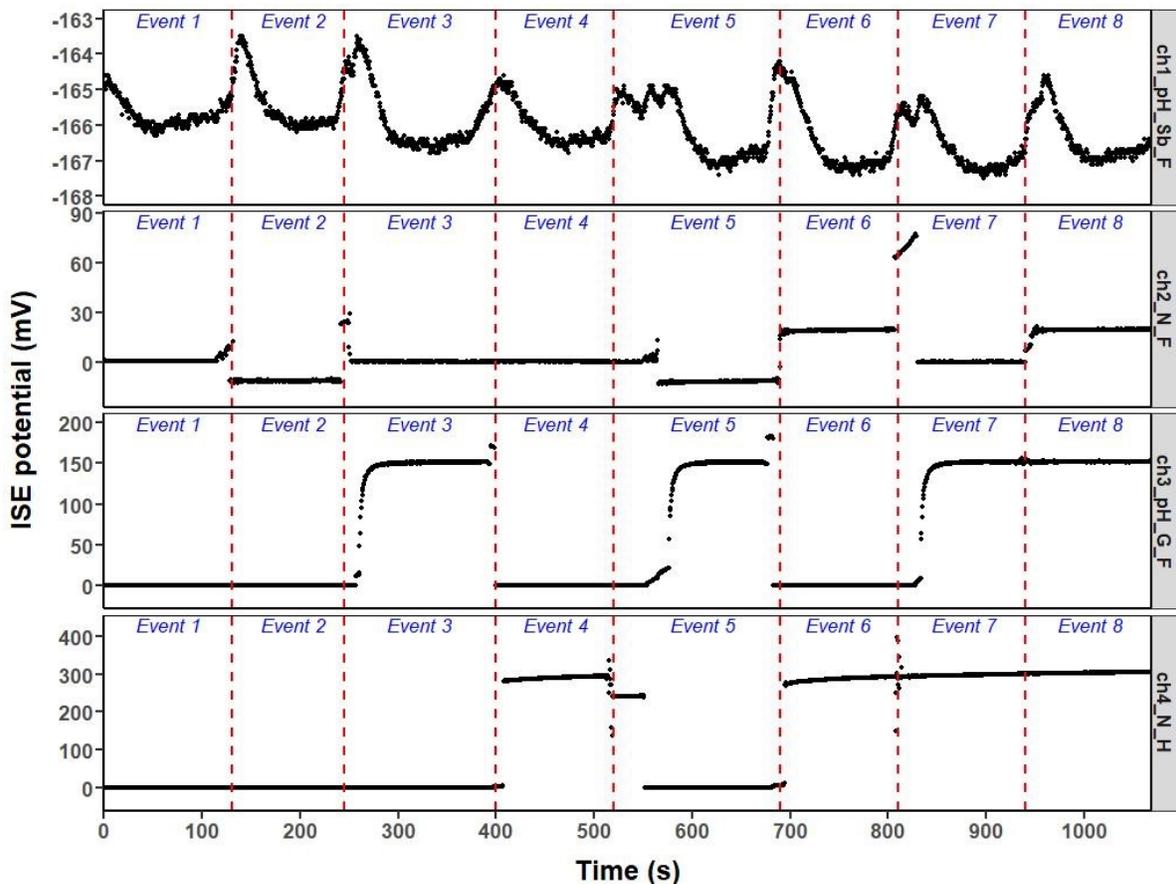


Figure 4-7. The first Tentacle (100 ms sampling rate) cross talk test result (F : full-cell combination ISE, H : half-cell mono ISE)

The problem with the ground loop DAQ was also experienced by Sethuramasamyraja et al. (2008). They used two half-cell potassium and nitrate ISEs with a shared reference from a full-cell glass pH combination. This caused the ISE to provide an unstable reading due to multiple reference/grounding which creates a ground loop error. Ground loops happen when two or more circuits have their own designated ground and they are connected to each other (Vasilescu, 2006). Since the ISE is a potentiometric sensor, there was a resistance difference between the half-cell mono and the full-cell combination nitrate ISE when they have the same reference. Thus, voltage noise appears and alters the ISEs reading as seen in at channel 2, event 6 (700 to 800 s).

Figure 4-8 represents the crosstalk and ground loop test result for the second Tentacle. There were no crosstalk and ground loop errors observed. Therefore, the second Tentacle was well isolated and appropriate for the DAQ.

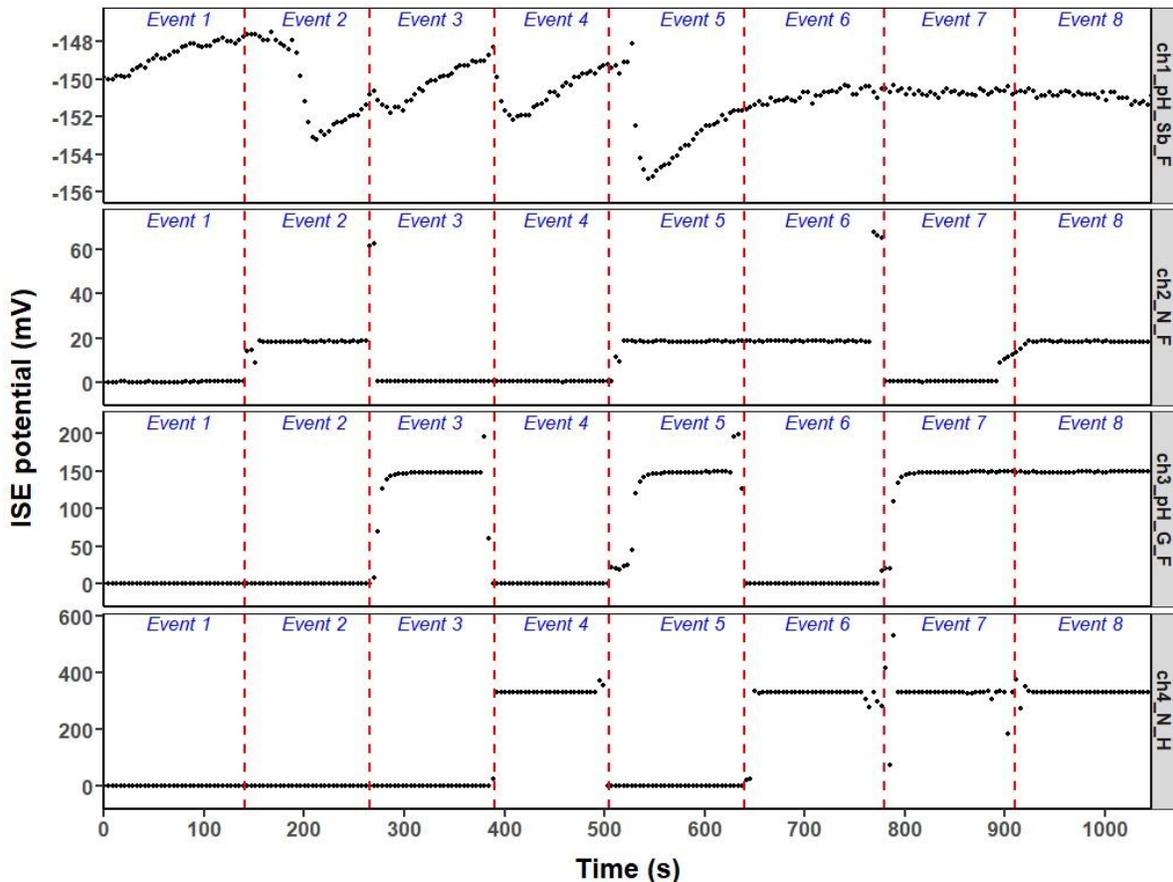


Figure 4-8. The second Tentacle (1000 ms sampling rate) channel 1 cross talk test result (F : full-cell combination ISE, H : half-cell mono ISE)

4.3.3. Manual Soil Analyzer Field Operation

The manual sampler was able to collect and analyze the soil samples. The footstep and the variable depth attachment provided a firm position during sampling. Compared to the existing benchtop soil analyzer (Li et al., 2019; Miao et al., 2008), the manual soil analyzer offers simpler soil sensing platform. Additionally, the manual soil analyzer able to do variable depth and multiple ISEs measurements for the topsoil chemical status profiling. The ISEs holder design was able to ease the operator adjusting the ISEs to properly contact the soil sample. Also, it was able to reduce the ISEs breakage from excessive force used by the operator during the measurement process. The rinsing action successfully cleaned the ISEs. Depending on the soil conditions, usually the operator only needed to operate the manual piston pump once to get a decent ISEs cleaning result.

4.3.4. DAQ Performance

The ISEs performance is summarized in Table 4-3. Field calibration for the first and second experiments are given in Figure 4-9 and Figure 4-10, respectively. In both experiments, antimony pH and nitrate ISEs are able to characterize both soil pH and nitrate concentrations. In the first experiment, the glass and antimony pH had an acceptable R^2 of 0.67 and 0.53 with RMSE of 0.17 pH, respectively. Better prediction were found with the nitrate ISE with R^2 of 0.83 (Figure 4-9c) with RMSE of 0.29 pNO₃. Comparing Figure 4-9a and b, the glass pH ISE was slightly underestimating the true soil pH whereas the contrary happened for the antimony pH ISE. This is because the ISEs are placed according to depth. The outer most ISE measures the deepest soil sample while the inner ISE measures the shallower soil profile.

Table 4-3. Manual soil analyzer ISEs performance

ISE	Reference	Nernst Slope (mV/decade)	Precision (pX)	Accuracy (pX)	n Sample
pH, glass combination*	-	-49.6	0.17	0.23	11
pH, antimony combination*	-	-49.6	0.17	0.30	11
NO ₃ , mono*	pH Sb	54.8	0.29	0.31	11
pH, antimony combination**	-	-47.9	0.20	0.91	9
pH, epoxy combination**	-	-53.8	0.36	0.34	8
NO ₃ , combination**	-	53.3	0.21	0.31	9

*) the first experiment 2018; **) the second experiment 2019; pX = -log [X]

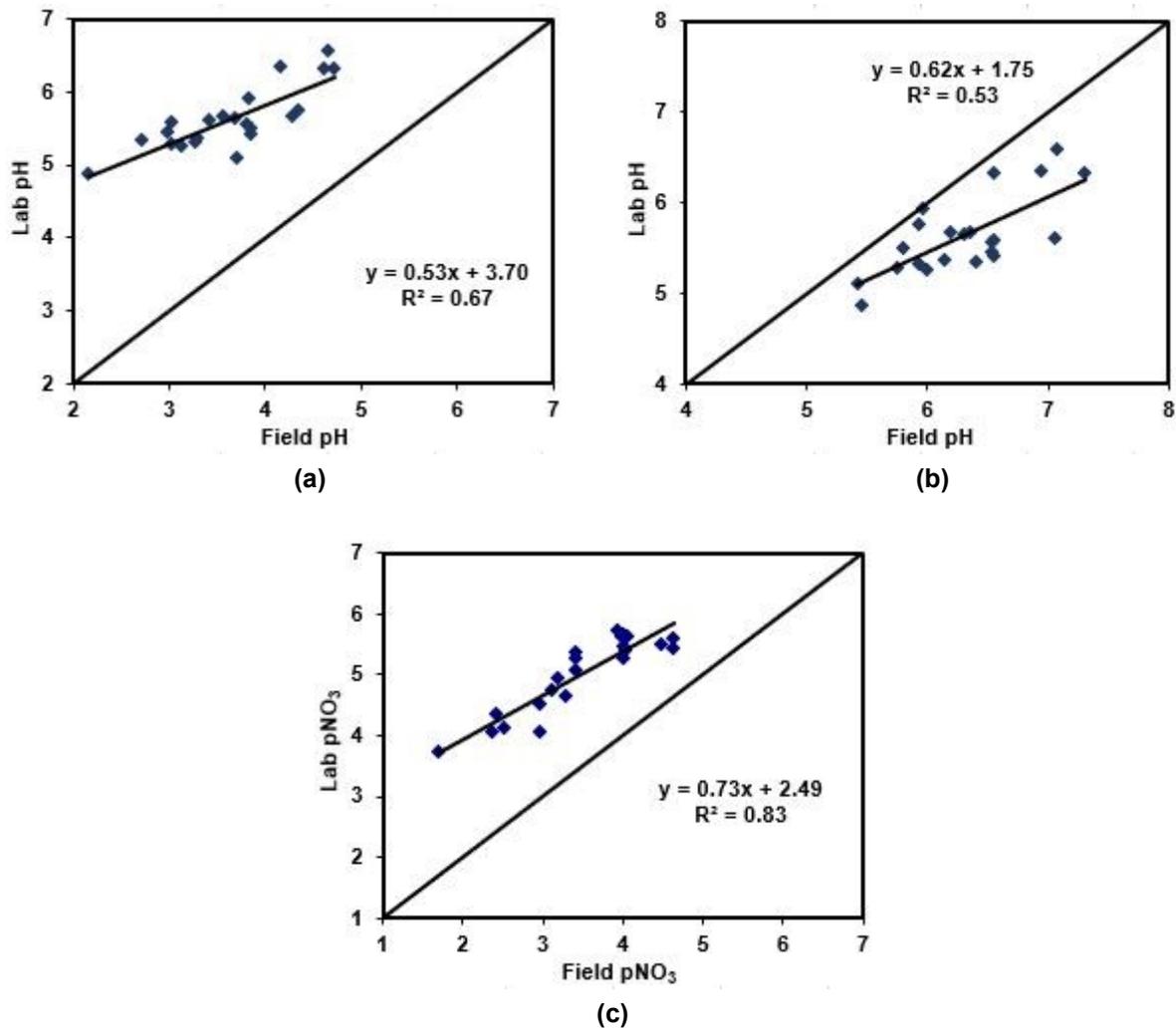


Figure 4-9. The first experiment ISEs field calibration
(a: pH glass b: pH Sb, c: nitrate PVC-Y360)

In the second experiment, although there was an accumulation of basic soil pH data (> pH 7), the epoxy pH ISE was able to characterize soil pH satisfactorily with R^2 of 0.88 (Figure 4-10b) and RMSE of 0.36 pH. However, the antimony pH ISE did not show good soil pH calibration (Figure 4-10a) presumably attributed to poor contact with the soil sample. With a multiple channel DAQ, an additional ISE can be used to provide cross validation on the soil chemical measurements as well as backup. Better results were achieved from the nitrate ISE with R^2 of 0.84 (Figure 4-10c) and RMSE of 0.21 pNO₃.

In both experiments, it can be inferred that nitrate extraction shifted the nitrate ISE calibration upward as more nitrate ions were extracted in the measured solution. Nevertheless, the standard error of the nitrate ISEs were comparable to what found by

Price et al. (2003). The arrangement of ISEs also affected the calibration result, especially for pH ISEs as they were placed to measure the soil sample at the maximum soil sampler depth of 20 to 25 cm. With this ISE's positioning, it might negatively affect the contact between the ISEs and the soil sample as soil becomes more compacted with depth. Nonetheless, the standard error of pH ISEs were acceptable compared to typical DSM (Adamchuk et al., 2005).

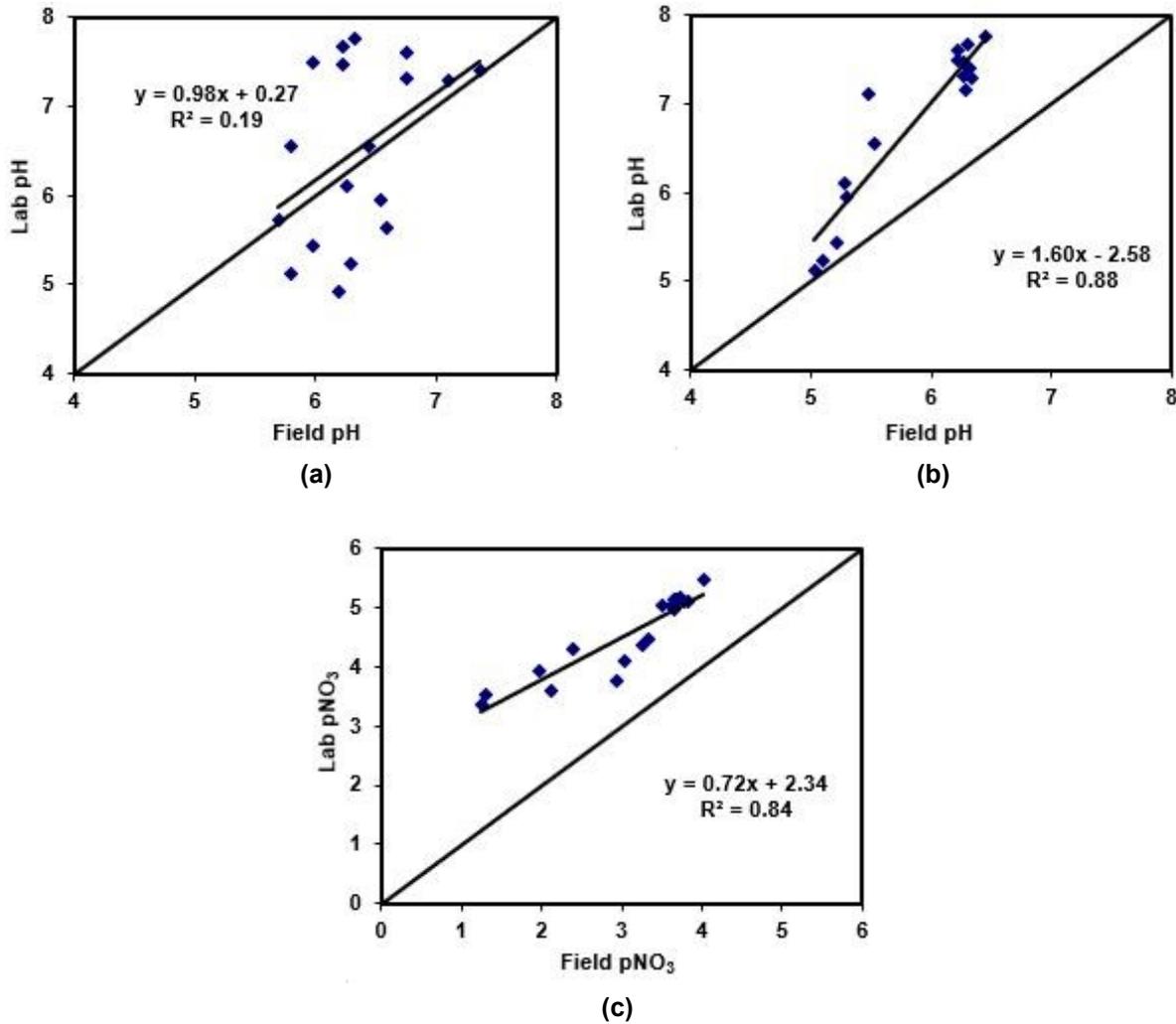


Figure 4-10. The second experiment ISEs field calibration
(a: pH Sb, b: pH epoxy dome, c: nitrate PVC)

4.4. Conclusion

The developed manual soil analyzer was convenient to use, offered good ISEs data acquisition and effectively cleaned the ISEs after sampling. The 1-s sampling rate Tentacle was chosen as the DAQ platform because it has better noise, cross talk and ground loop protection. In both field tests, the pH ISE performed satisfactorily when predicting soil pH with R^2 between 0.53 to 0.88 with RMSE ranging from 0.17 to 0.36 pH. The nitrate ISE performed better in both experiments with R^2 between 0.83 and 0.84 with RMSE of 0.21 to 0.29 pNO₃. Overall, the new manual soil sensing system would provide a portable alternative for depicting multiple soil chemical variability in smaller fields.

Connection to Chapter 5

Apparent soil electrical conductivity (EC_a) maps provide information for determining suitable sampling locations for various soil property measurements (Heil and Schmidhalter, 2017). By using EC_a maps to select the appropriate soil sampling locations, this directed sampling supports the soil sensing platform operations developed in the two previous chapters. Furthermore, this information can be used to delineate management zones with additional soil analyses information such as soil pH and nitrate gathered by soil sensing platform (Fridgen et al., 2004; Corwin et al., 2008). However, the interpretation of the EC_a measurements is often site-specific. The following chapter discusses the development of soil EC_a inversion using a fixed slice Brute-Force approach as a way to solve the EC_a interpretation problem. The Brute-Force method was chosen due to its simplicity and straight forward calculation utilizing the LIN assumption of the EMI sensor. The results from the Brute-Force EC_a inversion are a two-layer soil model at different depths together with their corresponding soil EC_a values. This research was published in the ISPA conference 2018 in Montreal, Canada and CSBE conference 2018 in Guelph, Canada.

Leksono, E., V. Adamchuk, W. Ji, and M. Leclerc. 2018. Development of a soil EC_a inversion algorithm for topsoil depth characterization. In: Proceedings of the 14th International Conference on Precision Agriculture, Montreal, Quebec, Canada. 24-27 June 2018. International Society of Precision Agriculture (published on-line at <http://www.ispag.org>, 11 pages).

Leksono, E., V. Adamchuk, H. Akbarzadeh, M. Leclerc, and R. Buelvas. 2018. Development of a subsoil manual electrical conductivity probe. Paper No. 18-201. Orleans, Ontario: CSBE.

5. Development of Brute-Force Soil EC_a Inversion Algorithm

Abstract

Electromagnetic induction (EMI) proximal soil sensor systems, such as the DUALEM-21S (Dualem, Inc. Milton, Ontario, Canada), are widely used electromagnetic induction (EMI) sensors for delivering rapid information about apparent soil electrical conductivity (EC_a). Depending on the instrument configuration, soil EC_a readings correspond to different depths of investigation. The interpretation of the EC_a measurements is often site-specific. Inversion is required to explore specific depths. This inversion process is an “ill-posed” problem which might lead to non-existing, or non-unique solutions. Commonly, a complicated regularization method is chosen to tackle this problem. In this chapter, a simple exhaustive Brute-Force method was developed to characterize soil layering depths and their corresponding EC_a values. High density EC_a data need to be filtered to assure that the data are in the low induction number (LIN) range which is what the Brute-Force inversion is based on. A two-layer soil EC_a model was used to represent both shallow and deeper soil layers. Brute-Force EC_a inversion will depict the depth of each layer and its corresponding EC_a value. From the high density DUALEM-21S input data, the Brute-Force algorithm was successfully converged to the minimum root mean squared error (RMSE) for each depth increment. The software’s GUI was intuitive and provided an up to date progress of the calculations. This algorithm has been tested successfully to determine the shallow and deeper soil EC_a values together with their layering depth on the 25-ha field near Naperville, Quebec, Canada. From those maps, important soil profile parameters can be determined and soil sampling scheme can be planned effectively for the soil sensing platform operation.

Keywords. electromagnetic induction, soil EC_a, inversion, soil layering depth.

5.1. Introduction

Proximal soil sensors, such as electromagnetic induction (EMI) sensors, can deliver decent spatial and temporal information about soil. EMI sensors measure apparent soil electrical conductivity (EC_a) and have become a common way to rapidly characterize soil heterogeneity (Corwin, 2008). DUALEM-21S (Duaem, Inc., Milton, Ontario, Canada) is a popular example of an EMI sensor used in precision agriculture (Saey et al., 2009). The DUALEM-21S is a dipole configuration EMI sensor with a fixed working frequency of 9 kHz. It has one vertical transmitter (T_x) coil with two sets of receiver (R_x) coils spaced 1 and 2 m for the horizontal coplanar orientation (HCP) and 1.1 and 2.1 m for the perpendicular coil orientation (PRP).

The EMI sensors provide valuable information about changes in EC_a magnitude with depth which is valuable for investigating the depth of homogeneous soil layering and its corresponding EC_a value. The interpretation of soil EC_a readings for different soil properties requires special skillsets since conduction in soil can be affected by various factors (Bronson et al., 2005; McNeill, 1980a). The goal of EMI sensor data interpretation is to determine field anomaly or target soil properties as well as the EC_a layering characteristics of a field through inverse modeling (Daniels et al., 2008). However, the inversion process from EC_a data to the corresponding soil depth and EC_a values is an ill-posed problem (Zhdanov, 2015) which is rather complicated and not suitable for typical agricultural applications.

There are various methods in inverse modelling, but most popular are the finite element method and the fixed slice cumulative depth response approach. The finite element method has been used extensively to investigate various soil properties using different EMI sensors with favorable results (Santos, 2004; Santos et al., 2010a; Triantafyllis et al., 2011; Huang et al., 2016). Unfortunately, it requires advanced user skills in data preparation, initial EC_a model setup and inversion results interpretation. The fixed slice cumulative depth response approach has been used successfully for subsoil characterization (Saey et al., 2008; De Smedt et al., 2013; Saey et al., 2012a). This inversion method is preferable due to its simplicity and its straightforward calculation

utilizing the low induction number (LIN) assumption of the EMI sensor. Therefore, it is valuable for the farmer's use.

In this research, an exhaustive Brute-Force method was incorporated into the fixed slice cumulative response approach to characterize soil layering depths and their corresponding EC_a values. Additionally, the inversion calculation was bounded in LIN approximation without considering other factors, such as wave scattering and propagation effects throughout different materials. Using the LIN calculation is essential to simplify the inversion algorithm and importantly, it is the basis used by most EMI sensors to estimate EC_a values.

The high density EC_a data source from the DUALEM-21S EMI sensor has four measurement modes. Hence, it can be used to generate up to four unknowns characterizing the soil profile. Thus, a two-layer soil EC_a model was sufficient to depict the depth of the shallow soil layer and its corresponding EC_a value. The two-layer model represents a shallow and deeper soil depth with $\sigma_{<d}$ and $\sigma_{>d}$ as soil shallow and deep EC_a , respectively. These arrangements were considered to be sufficient to determine the depth of the shallow soil layer (muck soil) over clay subsoil in a Quebec vegetable production farm.

5.2. Materials and Methods

5.2.1. DUALEM-21S Mapping

DUALEM-21S mapping was performed on a 25-ha field, predominantly muck soil, located at Napierville, Quebec, Canada (Figure 5-1). As a typical field setting, the sensor was towed behind a vehicle and sampling was performed in a transect arrangement. Three blocks of DUALEM-21S data streams were received sequentially by the PC; these consisted of timestamped EC_a readings from all coil arrangements, instrument operational and positional information (Appendix 9.5). A LabVIEW (National Instruments, Austin, Texas, USA) software was used to receive and parse the incoming data stream. The parsed data were stored in the PC in a *.txt file with corresponding data headers. The DUALEM-21S sampling rate was set at 1 Hz resulting in an approximate 5 m separation distance between records (mean of 10 consecutive measurements). The distance between transects was set to 10 m.



Figure 5-1. DUALEM 21-S mapping location

5.2.2. DUALEM-21S Data Filtering

Reliable EMI data are valuable for accurate interpretation and for further processing, such as soil EC_a inversion (Christiansen et al., 2016). Various steps are involved in DUALEM-21S data filtering, such as removing outliers (positional, instrumental and

environmental noise), smoothing, and data decimation to provide clean, usable information. Importantly, the removal of beyond the low induction number (LIN) measurement is the pivotal step in the DUALEM-21S data filtering process (Figure 5-2). Matlab R2018b (MathWorks Inc. Natick, Massachusetts, USA) was used as a platform to develop the filtering algorithm and the graphical user interface (GUI) (Figure 5-3).

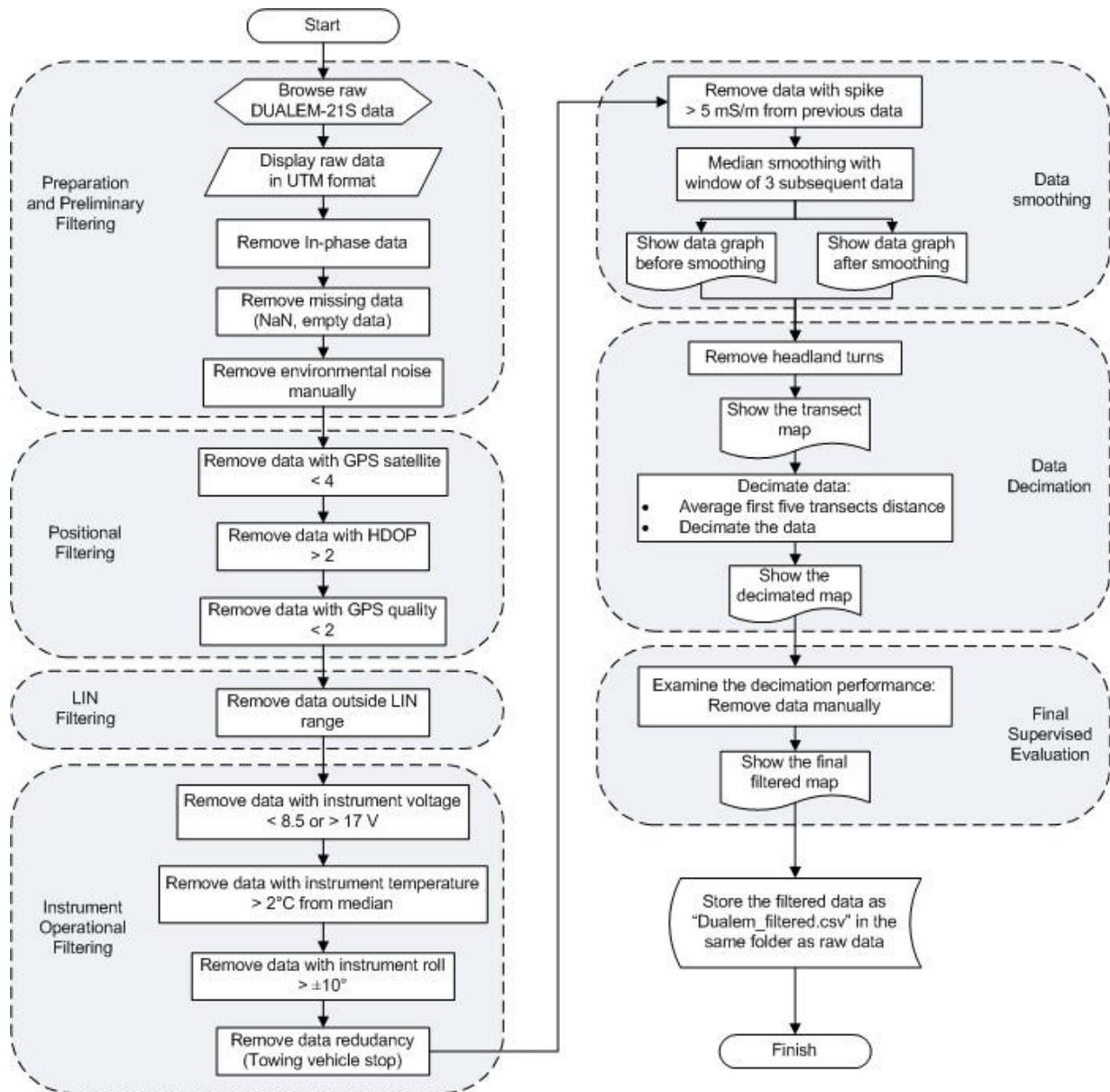


Figure 5-2. DUALEM-21S filtering hierarchy

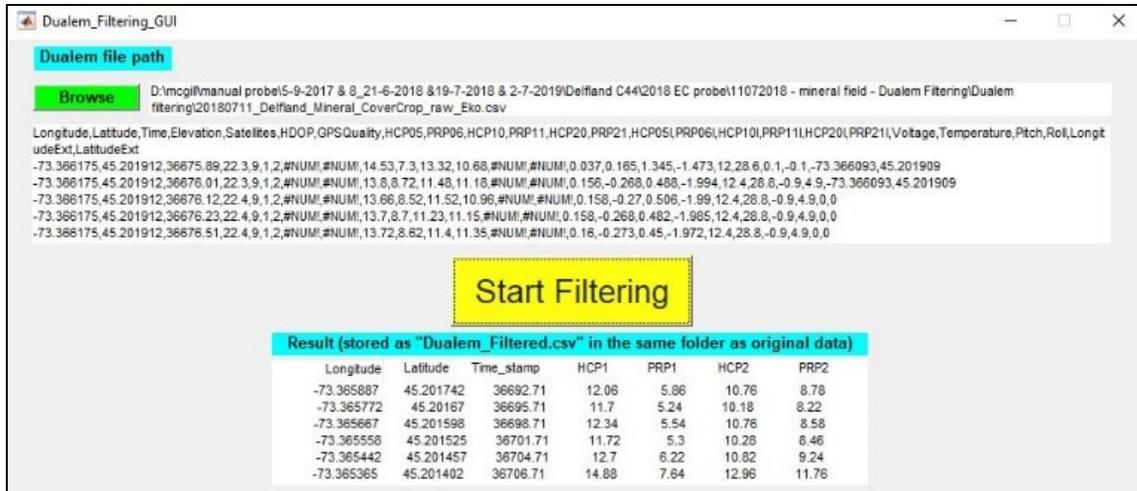


Figure 5-3. DUALEM-21S filtering GUI

The preparation and preliminary filtering involved the removal of unused sensor data headers, missing data and environmental noise. The in-phase reading from each receiver coil was removed. Additionally, missing data were cleaned to compensate for random environmental noise which can stop, or change, the data format received, thus, resulting in an incomplete data stream. Supervised data filtering is necessary in this step to enable users to thoroughly examine the sensor's reading nearby the suspected noise location and to selectively delete the unwanted reading.

The positional data from the DUALEM-21S internal GPS are filtered out if it has < 4 GPS satellite contributing to the position estimation. Furthermore, data with Horizontal Dilution of Precision (HDOP) > 2 attributes are removed. The internal DUALEM-21S GPS support WAAS (Wide Area Augmentation System) and represented with GPS quality of 2. As the typical field surveying takes place in North America where there is WAAS support, less than 2 GPS quality needed to be filtered to provide reliable georeferenced data.

The low induction number (LIN) filtering removed the negative quadrature readings as the typical field EC_a value was considered positive and the sensor was assumed to be well calibrated. One of the EMI sensor's characteristics is that it has a distinct skin depth and induction number for all Tx/Rx arrangements. Skin depth is the depth where the primary field is attenuated to 1/e (36.8%), while the induction number is the ratio of inter

coil spacing (Tx/Rx distance) to skin depth (McNeill, 1980b) and represented by the following equation:

$$\beta = \left(\frac{2}{\omega\mu\sigma} \right)^{\frac{1}{2}} s \quad (5-1)$$

where β is induction number, ω is equal to $2\pi f$, f is the EMI sensor transmitter frequency (Hz), μ is the magnetic permeability of conductor (H m^{-1}), σ is the electrical conductivity of the material (S m^{-1}) and s is the inter coil spacing (m).

According to DUALEM-21S documentation (DUALEM, 2018), LIN range for PRP and HCP orientation is less than 0.50 and 0.16 induction number respectively. With DUALEM-21S Tx frequency of 9 kHz, for HCP 1 and 2 m (further called HCP1 and HCP2 respectively), sensor value of more than 720.3 and 179.4 mS m^{-1} were removed. While for PRP 1.1 and 2.1 m (also called PRP1 and PRP2, respectively), sensor values of more than 5807.8 and 1593.7 mS m^{-1} were removed.

DUALEM-21S works on 8.5 to 17 VDC, so any data that has an instrument voltage outside that range is removed. Internal instrument temperature also becomes a consideration as it contributes to sensor drift (DUALEM, 2014; Sudduth et al., 2013; Mester et al., 2011). Thus, data with more than 2°C difference from the temperature median of the whole data are filtered. Furthermore, data with an instrument roll more than 10° from the center position are removed as it is considered to exceed 2% quadrature deviation (Dabas et al., 2016). Data redundancy when the DUALEM's towing vehicle stopped were also removed.

There are several alternatives to filtering and smoothing the EMI sensor data, such as Canny or Gaussian filtering (Saey et al., 2012a). In our case, two steps were involved. A localized data spike with more than 5 mS m^{-1} difference from the adjacent data were removed. Furthermore, median filtering with a window of 3 subsequent data was employed to smooth and improve the signal to noise ratio for each Tx/Rx configuration reading. Median filtering was considered sufficient to provide decent smoothing for a typical field survey rather than using a grid, circle or other neighborhood filtering (De Smedt et al., 2013).

The headland data when the towing vehicle makes a turn to the next adjacent transect were removed. The data was filtered if it deviated more than 15° from the three previous data point locations. This arrangement was considered sufficient to compensate for the operator's steering movements while driving straight. Additionally, to provide equal spatial distribution, data needs to be decimated. The decimation method was done by taking the average distance between the first four transects; then all data were decimated based on this distance. To examine the decimation result, a pop-up graph appears for supervised filtering by the user. The filtered DUALEM-21S data were saved in the same folder as the raw data input.

5.2.3. DUALEM-21S Response Function

DUALEM-21S is an active EMI sensor that creates a time varying primary electromagnetic (EM) field through its transmitter (T_x) coil. The surrounding conductive bodies that receive this primary field will become electromagnetically induced and an eddy current will be created. This current will create its own secondary EM field that opposes the primary EM field direction. Depending on the characteristics of the conductive materials, the primary EM field will experience a phase, amplitude, and direction change. In addition, the receiver (R_x) coil receives both the in-phase EM signals from the T_x coil and the out-of-phase quadrature signal from the conductive body and interprets them as apparent soil electrical conductivity (EC_a) in S m⁻¹ (Daniels et al., 2008).

EMI sensors measure soil EC_a under the assumption of linearity between measured EC_a and the true homogeneous halfspace conductivity. The linear relationship only holds at the low induction number. The induction number is the ratio of inter coil spacing to skin depth. The depth where the primary field is attenuated to 1/e (36.8%) is called skin depth. Within this range, soil EC_a can be described as:

$$EC_a = \frac{4}{\omega\mu_0 s^2} \frac{H_s}{H_p} \text{ (S m}^{-1}\text{)} \quad (5-2)$$

where ω is $2\pi f$ (s⁻¹), μ_0 is the permeability of free space ($1.25663706 \times 10^{-6}$ m kg s⁻² A⁻²), s is the primary to secondary coil (inter coil) separation (m), H is the electromagnetic field with subscript s and p refer to secondary and primary field respectively (A m⁻¹).

Soil is not uniform and hence, it is comprised of various permeability levels (Patitz et al., 1995). Therefore, EC_a interpretation needs special training and often requires other sensors to validate the EC_a measurements. Under the low induction number (LIN) assumption, the relative (ϕ) and cumulative (R) depth response function for vertical (v), and perpendicular (p) coils are the following:

for vertical dipole orientation (HCP),

$$\phi_v(z) = 4(z)(4z^2 + 1)^{-3/2} \quad (5-3)$$

$$R_v(z) = 1 - (4z^2 + 1)^{-1/2} \quad (5-4)$$

while for perpendicular dipole orientation (PRP),

$$\phi_p(z) = 2(4z^2 + 1)^{-3/2} \quad (5-5)$$

$$R_p(z) = 2z (4z^2 + 1)^{-1/2} \quad (5-6)$$

where z is normalized depth (soil depth divided by inter coil spacing). The response of the n -th soil layer to the cumulative EC_a (σ_a^c) can be described as:

$$\sigma_a^c = \sigma_1 R_{(z_1)} + \sum_{i=2}^{n-1} \sigma_i [R_{(z_i)} - R_{(z_{i-1})}] + \sigma_n [1 - R_{(z_{n-1})}] \quad (5-7)$$

where σ_1 is a zero (air) conductivity since EMI sensor might be used at different height to particularly examine a specific depth of interest. Following are the graphs of relative and cumulative depth response functions:

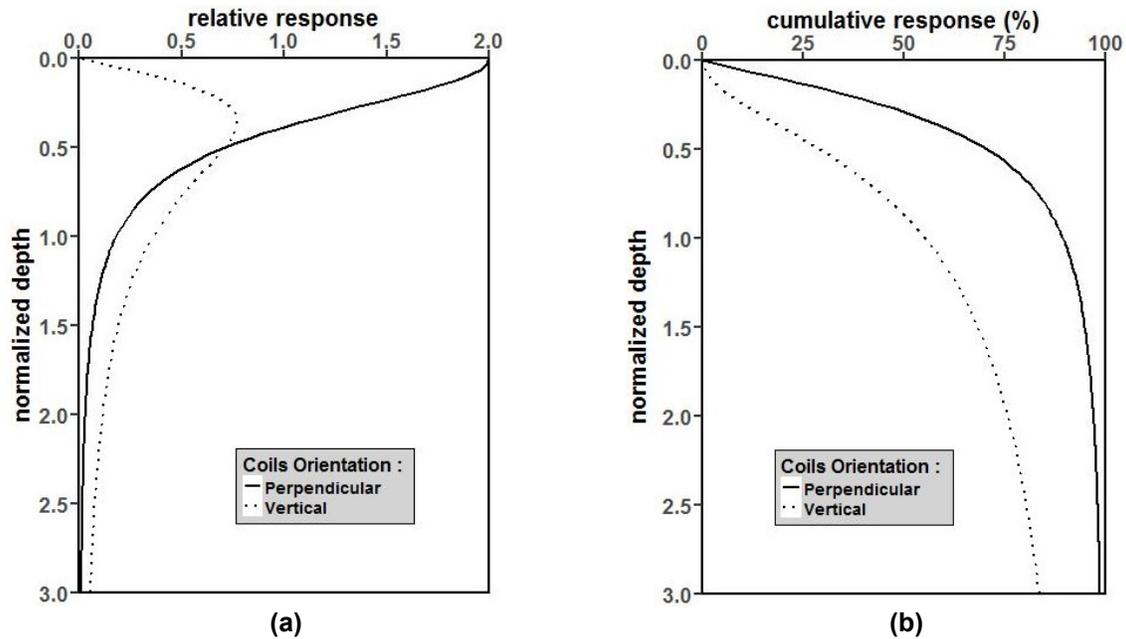


Figure 5-4. Dualem 21-S response function.
 (a) relative and (b) cumulative response function

5.2.4. Brute-Force Inversion Software

The fixed slice cumulative depth response approach was selected as a base method to invert the soil EC_a measurement due to its simplicity and straight forward calculation. Matlab R2015b (MathWorks Inc. Natick, Massachusetts, USA) was used as the platform to develop the algorithm and its GUI. The two-layer model was used to represent shallow and deeper subsoil. The soil depth increment was set to 10 cm with a maximum depth of 150 cm. The modification from conventional methods relies on replacing Levenberg-Marquardt minimization with an exhaustive Brute-Force algorithm. Furthermore, the calculated EC_a model was iterated based on the modelled soil EC_a value instead of incrementing soil depth.

Figure 5-5 represents the GUI of the Brute-Force EC_a inversion software. There are two *.csv input files needed for the software to run: DUALEM-21S measurement data and cumulative depth response. For DUALEM-21S data, the column should be arranged to Longitude, Latitude, Timestamp, HCP1, PRP1 HCP2, and PRP2, respectively. While for the cumulative depth response, the data column should be arranged to Depth, PRP1, PRP2, HCP1, and HCP2 respectively. After successfully inputting the data, the GUI will print the source data file path and the five topmost data values.

Users can choose the data quantity to be processed by filling the “From” and “To” boxes (e.g. filling “From” with “500” and “To” with “1000” will invert the EC_a data from measurement point number 500 to 1000). This option allows the user to have multiple inversion software open at the same time. Also, it will allow users to specifically invert the EC_a measurements at a specific transect of interest. The user can monitor or cancel the inversion process anytime through the progress bar splash screen. Furthermore, a completion process screen will prompt the user if the inversion is finished. The inversion result is saved into the *.xlsx file format and it is stored in the same folder as the initial DUALEM-21S measurement data.

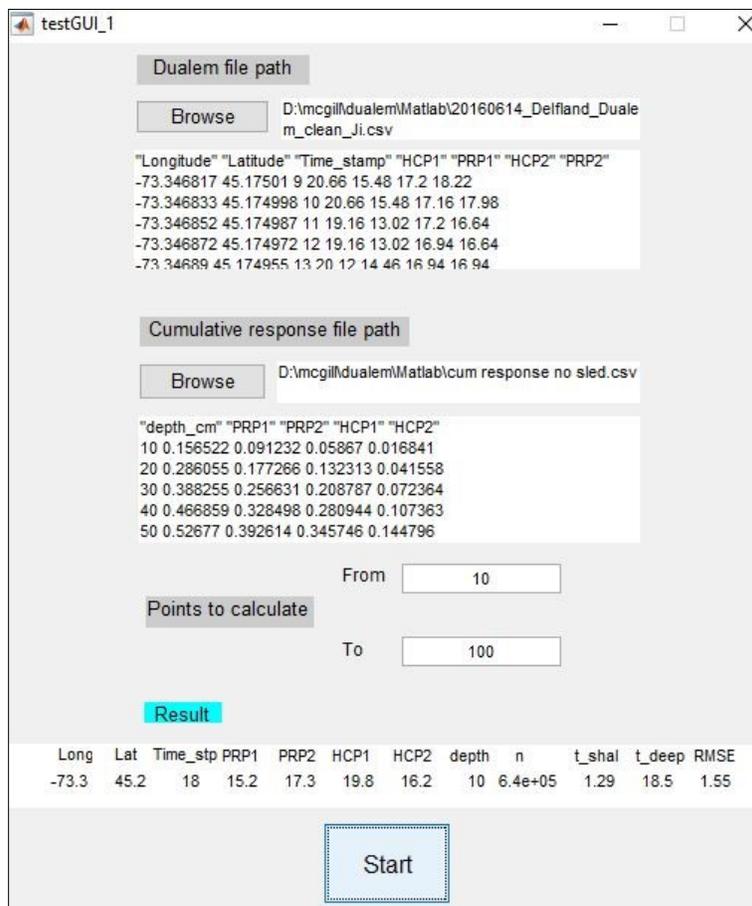


Figure 5-5. The Brute-Force inversion GUI

The Brute-Force inversion algorithm can be described as follows: assume that at a specific location the algorithm needs to calculate the soil EC_a at the first 10 cm depth (d = 10 cm) and below. The base setup matrices are:

$$\sigma_a^{cd} \times D = \sigma_a^{calc} \quad (5-8)$$

with,

$$\sigma_a^{cd} = \begin{bmatrix} R_{<d}^{P1} & R_{>d}^{P1} \\ R_{<d}^{P2} & R_{>d}^{P2} \\ R_{<d}^{H1} & R_{>d}^{H1} \\ R_{<d}^{H2} & R_{>d}^{H2} \end{bmatrix}, \quad D = \begin{bmatrix} \sigma_{<d} \\ \sigma_{>d} \end{bmatrix}, \quad \sigma_a^{calc} = \begin{bmatrix} PRP_1 \\ PRP_2 \\ HCP_1 \\ HCP_2 \end{bmatrix}$$

where σ_a^{cd} is 4 x 2 matrix consisting of DUALEM-21S cumulative EC_a response at shallow (<d) and deep (>d) soil depth with superscript P1 and P2 stand for PRP1 and PRP2 configuration respectively while H1 and H2 are for HCP1 and HCP2 configuration. D is 2 x 1 matrix consisting of the modelled shallow ($\sigma_{<d}$) and deep soil EC_a ($\sigma_{>d}$), σ_a^{calc} is 4 x 1 matrix consisting of the calculated EC_a value for all DUALEM-21S coil orientations.

The inversion process was started by defining each DUALEM-21S measurement mode with its top ($R_{<d}$) and deep ($R_{>d}$) soil cumulative depth response from Eq. (5-4) and (5-6) to form σ_a^{cd} matrix (e.g. with the instrument coil located at 12.7 cm above ground, the $R_{<10}^{P1}$ and $R_{>10}^{P1}$ would be 15.7 and 61.8%, respectively). Then, the forward calculation was performed to estimate the calculated EC_a (σ_a^{calc} matrix) from the DUALEM-21S cumulative response function (σ_a^{cd} matrix) and the modelled EC_a value (D matrix). The calculated EC_a value was then subtracted from the measured EC_a to obtain the error value by using the root mean squared error (RMSE) method. The maximum modelled EC_a value was set to 200 mS m⁻¹ as this is typical for a non-saline field (Staff, 2014), with a resolution of 0.2 mS m⁻¹. Therefore, overall, there are one million combinations of $\sigma_{<d}$ and $\sigma_{>d}$.

After all iterations, the cumulative depth response values of σ_a^{cd} matrix were changed at 10 cm depth increments after which this was followed by similar processes. Since the shallow soil layer is limited to a maximum 150 cm depth, we have fifteen sets of $\sigma_{<d}$ and

$\sigma_{>d}$. The appropriate depth combination solution would be the one that has the lowest misfit (RMSE) value. The inversion flowchart can be seen in Figure 5-6.

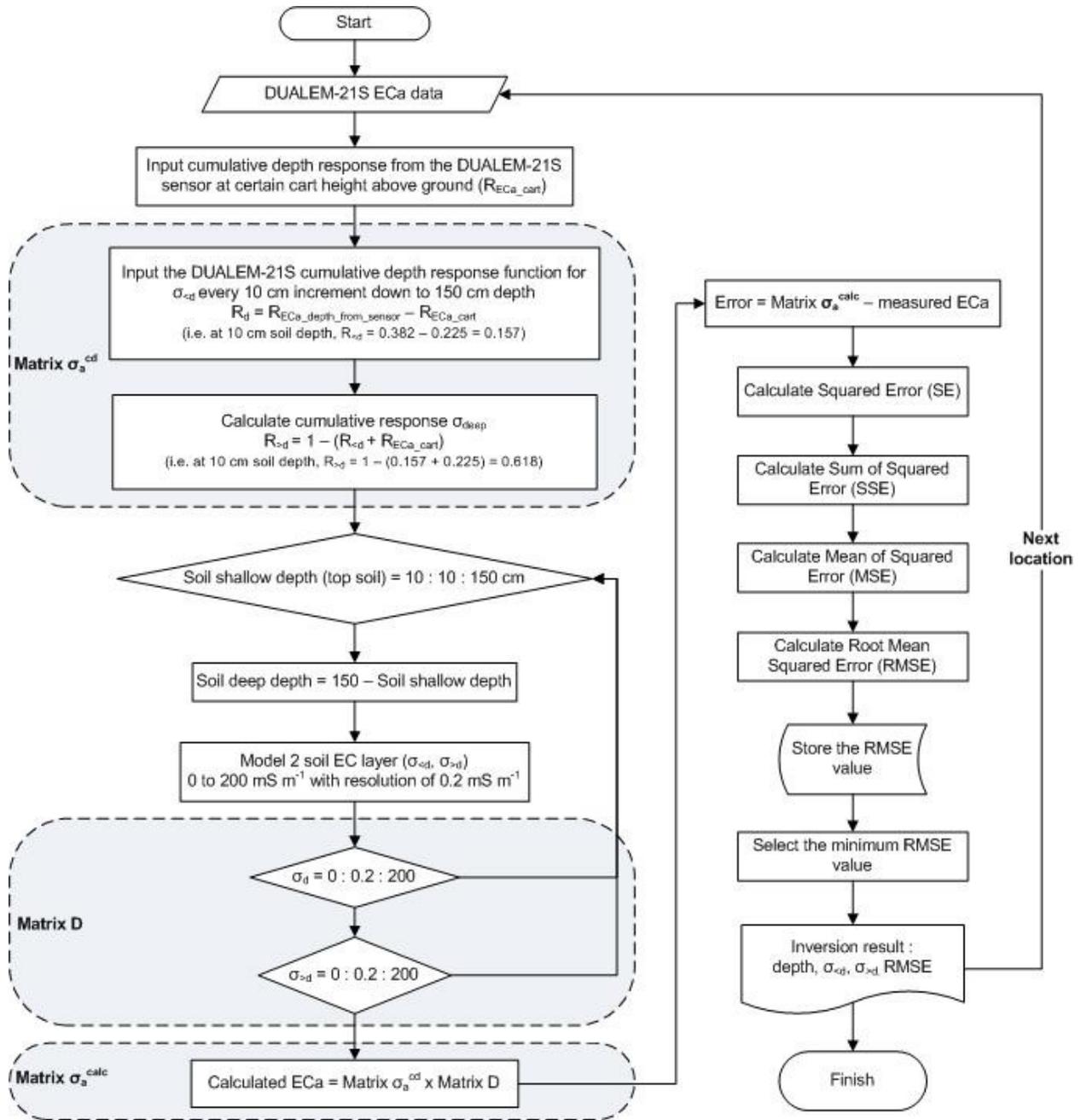


Figure 5-6. DUALEM-21S Brute-Force EC_a inversion flow chart

Before starting the inversion, all EC_a data points were filtered to achieve equal spatial resolution between points. Thus, 2828 DUALEM-21S data were used for the Brute-Force inversion from the initial 5655 data points. After the inversion was completed, the resulting shallow ($\sigma_{<d}$) and deep soil EC_a ($\sigma_{>d}$) were spatially interpolated using the Ordinary Kriging option in ArcMap 10.4.1 (ESRI, Redlands, California, USA). After the depth of the shallow soil EC_a model was determined, it could be used to calculate the volume of the field's shallow layer (Figure 5-7).

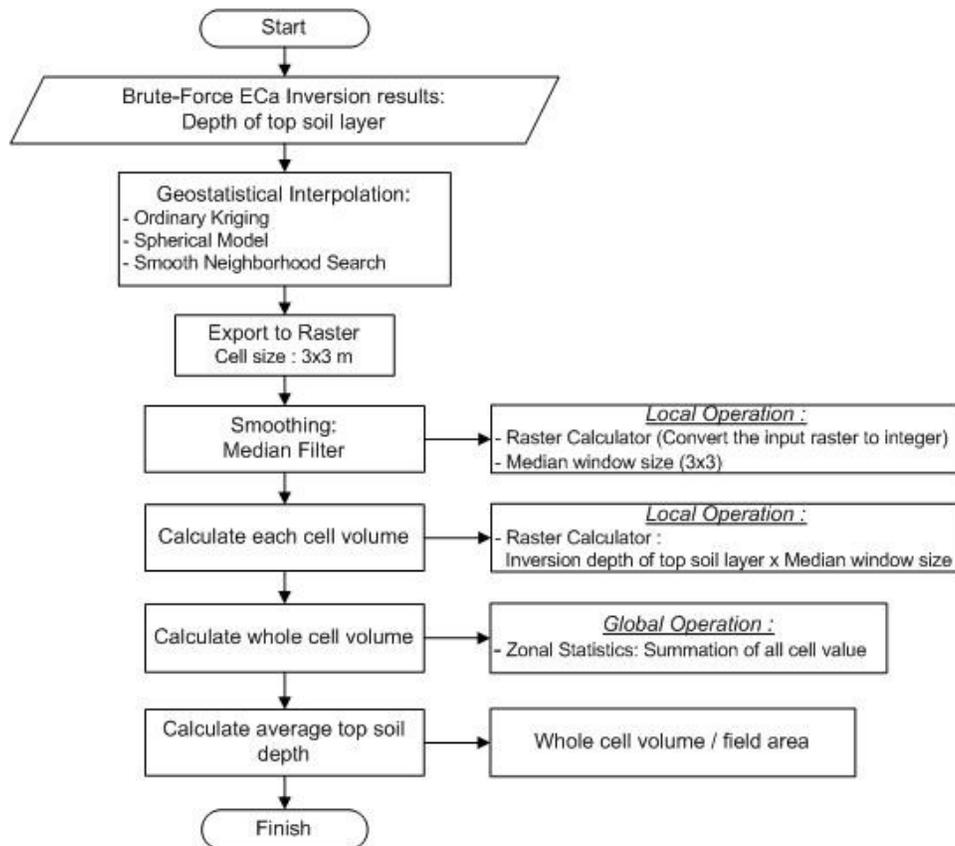


Figure 5-7. Procedure of shallow muck soil layer volume and average depth calculation from Brute-Force inversion result using ArcMap 10.4.1

5.2.5. Development of a Manual EC_a Probe

The outputs of the Brute-Force inversion are the EC_a value of the shallow and deep soil layer and the depth of the shallow layer. From this information, a series of maps corresponding to the Brute-Force inversion results can be generated. To examine the contribution of various soil properties and their stratification on the inversion result, information about the apparent soil electrical conductivity profile was needed.

Furthermore, various soil properties, such as organic matter content, texture and water content might affect the inverted EC_a output values and their corresponding depth. The additional EC_a profile information was important to understand the soil layering between the muck and the clay layers. One method used the extraction of soil cores, followed by examining the distinct soil characteristics by depth (Sudduth et al., 2013). However, this method was considered laborious, time consuming and expensive, especially with the Brute-Force EC_a inversion output which extends to a soil depth of 150 cm. Therefore, a manual subsoil EC_a probe needed to be developed.

The existing soil EC_a probe can only reach down to a depth of 60 cm (Figure 5-8). The new design was constrained by the need to use the existing Spectrum EC sensor tip (Figure 5-9) and it had to be able to penetrate non-saline muck soil down to 150 cm depth with the maximum 4 MPa cone index (Lowery and Morrison, 2002).



Figure 5-8. FieldScout EC_a 110 with 60 cm T-Handle EC Probe



Figure 5-9. Existing Spectrum EC sensor tip

To evaluate the required tube size and penetration force, a cone tip resistance equation (USDA, 2012) was used:

$$F_c = q_c \times A_c \text{ (N)} \quad (5-9)$$

where F_c is the axial force pushing down the probe tip (N), q_c is the cone tip resistance (Pa) and A_c is the tip surface area (m²).

For a maximum 4×10^6 Pa soil resistance and with 7.13×10^{-5} m² tip surface area from 9.825 mm diameter tube, the force needed to push down the EC_a probe 30° cone tip was 285.2 N. With the assumption of 70 kg operator weight, additional equipment to push the

EC_a probe down into the soil was not necessary. Furthermore, for simplicity purposes, a standard 180 cm 4130 Alloy tube was selected because it has the highest modulus elasticity and better weldability compared to other materials. Liquid silicon was used to waterproof the probe.

The new subsoil manual EC_a probe (Figure 5-10) was able to penetrate down to the 150 cm top layer depth and to directly measure the soil EC_a in non-saline soil with a maximum of 4 MPa soil penetration resistance (cone index). Figure 5-11 depicts the calibration of the subsoil manual EC_a probe before, and after, the field measurements.



Figure 5-10. Operation of the subsoil manual EC_a probe: (a) beginning of measurement at the soil surface, (b) end of measurement at 150 cm

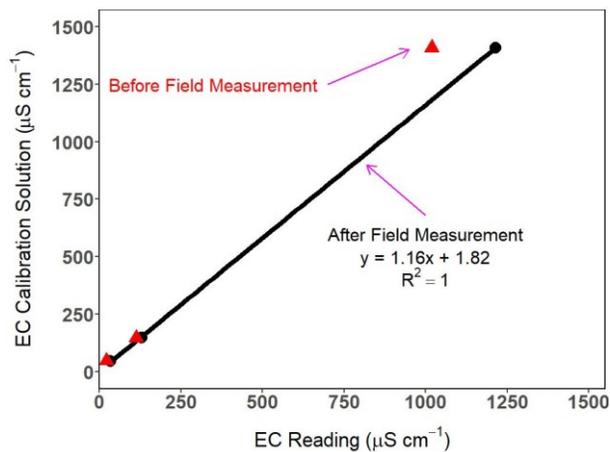


Figure 5-11. Subsoil Manual EC_a Probe calibration in 46, 147 and 1409 uS cm⁻¹ calibration solutions before and after field measurement

5.2.6. Soil Layering Model from Subsoil Electrical Conductivity Probe Data

A set of 30 sampling locations (Figure 5-12) was arranged to provide better coverage of both the shallow and deep EC_a inversion regions. The EC_a probe measurements were made at every 10 cm depth increments to depict the soil EC_a variations with depth. At each location, the EC_a profiles were arranged into two-layer soil EC_a models σ_{sp} and σ_{dp} for shallow and deeper soil layers, respectively. The appropriate soil layering depths were chosen based on the smallest RMSE value ($RMSE_{\sigma_{ECa\ model}}$). The following are the formula used to calculate shallow, deep EC_a model and their RMSE values, respectively:

$$\sigma_{sp(i)} = \frac{\sum_{i=1}^{N_d-1} \sigma_{ECa\ probe}}{i} \quad (5-10)$$

$$\sigma_{dp(i)} = \frac{\sum_{i=2}^{N_d} \sigma_{ECa\ probe}}{N_d - i + 1} \quad (5-11)$$

$$RMSE_{\sigma_{ECa\ model(i)}} = \sqrt{\frac{\sum_{i=1}^{N_d} (\sigma_{sp(i)} - \sigma_{ECa\ probe(j)})^2}{N_d}} + \sqrt{\frac{\sum_{i=1}^{N_d} (\sigma_{dp(i)} - \sigma_{ECa\ probe(j+1)})^2}{N_d}} \quad (5-12)$$

where N_d is the number of soil EC_a layers from the manual EC_a probe sampling, $\sigma_{ECa\ probe}$ is the measured soil EC_a at each soil depth increment, and $1 \leq j \leq N_d$. These profiles were compared with the Brute-Force inversion results. At each EC_a profiling location, the point-based Brute-Force inversion results were extracted from the Brute-Force inversion maps. The distinct EC_a transition layer was considered as the end boundary of the shallow muck soil layer.



Figure 5-12. Brute-Force EC_a inversion validation locations

5.3. Results and Discussion

5.3.1. Brute-Force Inversion Outputs

In one soil depth combination (e.g. shallow soil layer depth $d < 10$ cm and deep soil depth $d > 10$ cm), the inversion process results in one million $\sigma_{<d}$ (shallow) and $\sigma_{>d}$ (deep) EC_a values with their corresponding RMSE. Then, the inversion algorithm will select the minimum RMSE (red circle in Figure 5-13). After all depth combinations are inverted, the algorithm selects the minimum RMSE from the successive depth increments; in this illustration, it was 10 cm, which means that the depth of the shallow soil layer in this location was 10 cm. This process was repeated for each selected EC_a data point.

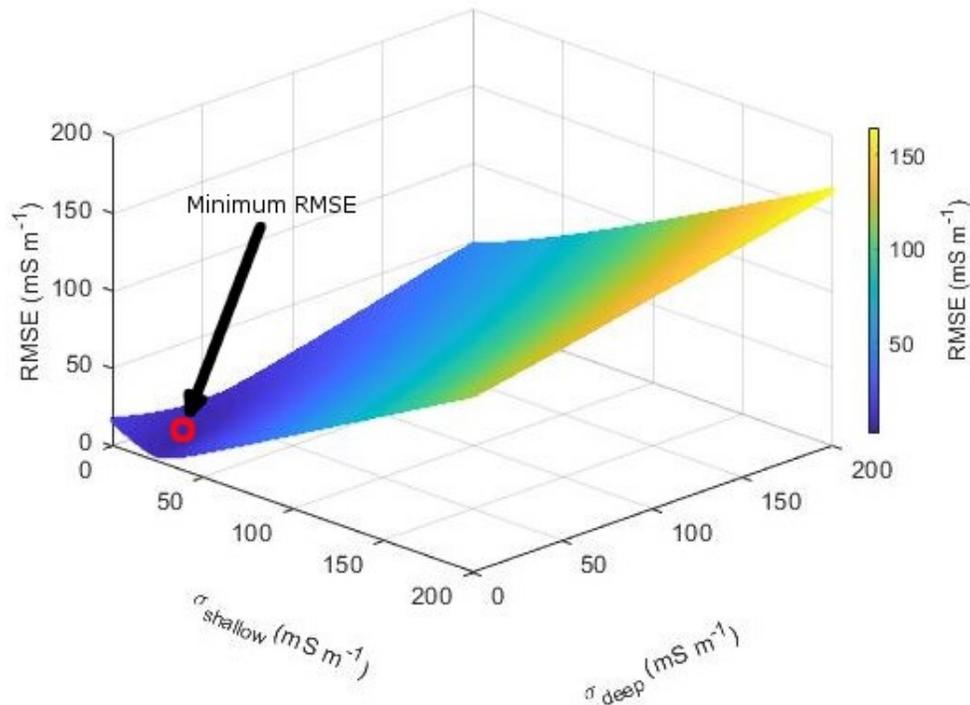


Figure 5-13. Three-dimensional graph of modelled soil EC_a $\sigma_{<d}$ (shallow), $\sigma_{>d}$ (deep) and their RMSE at $d = 10$ cm

The example of Brute-Force inversion results along the 725 m transect from the southwest area of the field is shown in Figure 5-14. It should be noted that there was no smoothing between the shallow and deep layers or between the adjacent transect points. Consequently, the inversion result may not represent the real soil EC_a gradient. As a remedy, spatial interpolation might be used to generate a smooth EC_a transition which will result in increased complexity and computation time. Figure 5-14 shows that the depth

of the shallow EC_a layers were between 10 and 150 cm. Furthermore, the inverted EC_a values in the shallow layer were lower than the layer below.

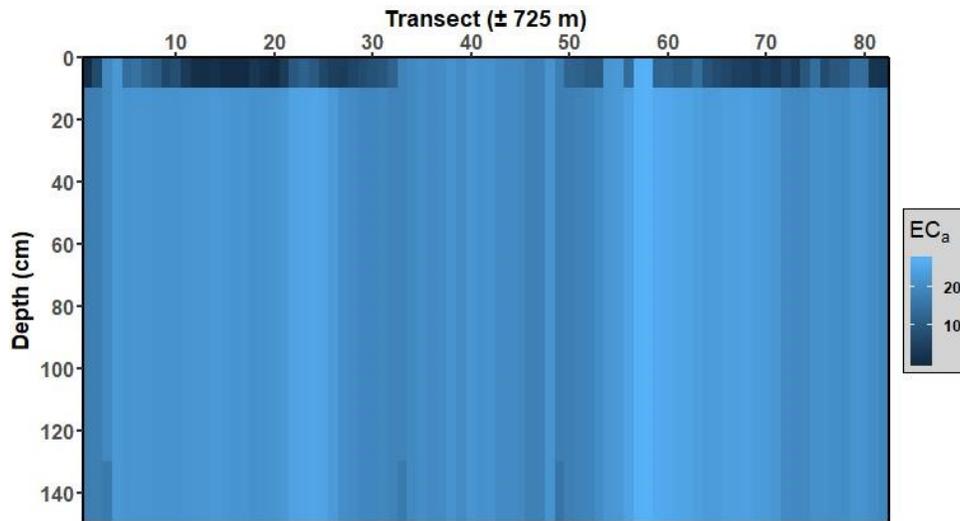


Figure 5-14. Example of Brute-Force inversion output along transect

5.3.2. EC_a Inversion Maps

One of the Brute-Force inversion results is a depth of shallow layer (Figure 5-15). From this map, the user can calculate the volume and the average depth of the shallow soil layer according to the procedure described in Figure 5-7. The volume of the shallow muck soil layer was found to be 222,470 m³ with an average depth of 89 cm. This valuable information about the volume and georeferenced depth of the muck soil layer provides farmers with options for adjusting their soil management practices. For example, this information is useful when deciding where to add extra organic matter or to select the appropriate precision irrigation practices.

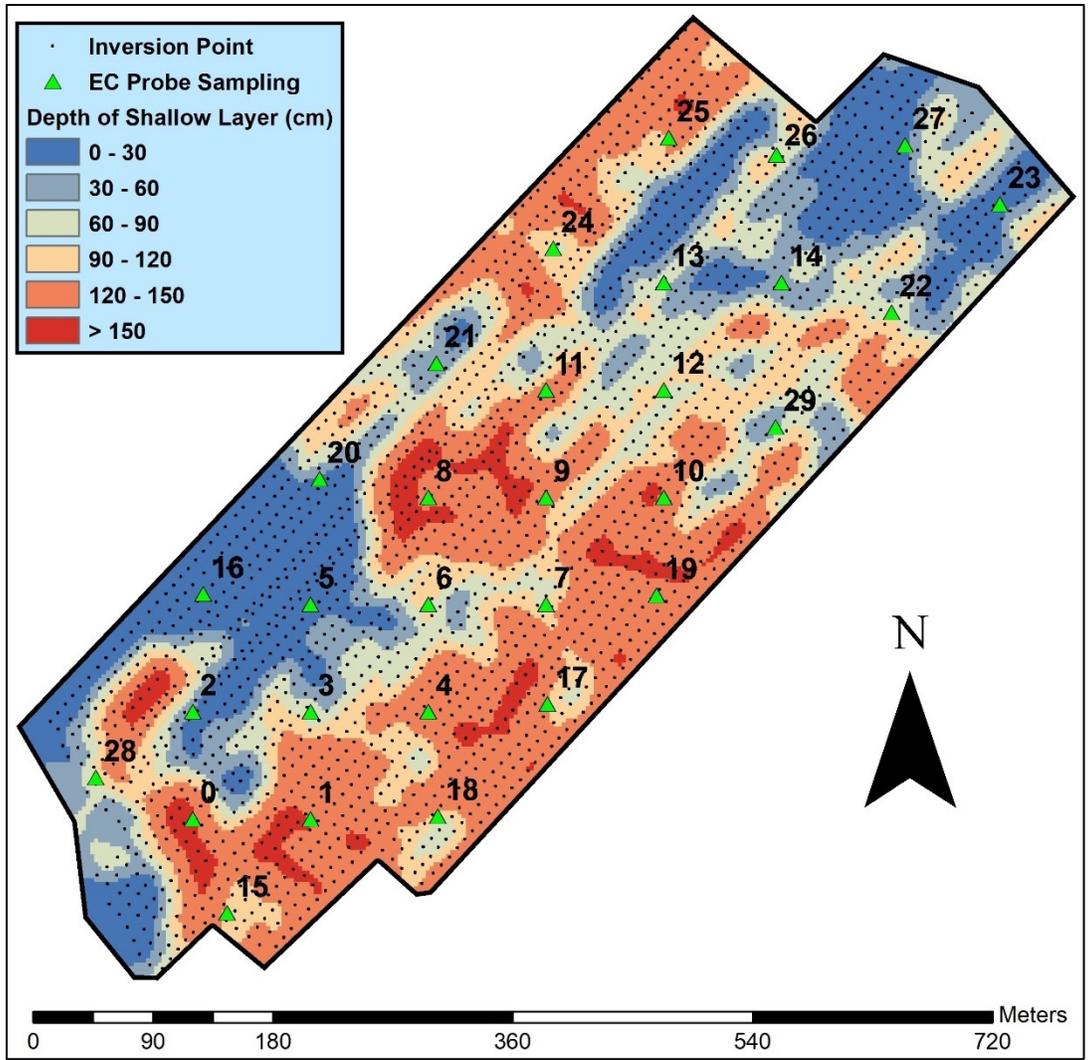


Figure 5-15. Depth of muck soil layer

The Brute-Force inversion result for shallow layer EC_a is shown in Figure 5-16. High soil EC_a at shallow depths is shown in the center, north-west, and east to south-east area of the field and this corresponds to the high organic content region of the soil. The high soil EC_a value in this region was in accordance with visual observations that the soil matches the visual characteristics of muck soil (Watson, 2007). Moreover, it was also noticed that in this area, the depth of the top muck soil layers was generally thick ranging from 90 to 150 cm (Figure 5-15). Soil organic matter content quantity is known to be positively correlated with soil water holding capacity (Hudson, 1994) and soil conduction is affected proportionally by soil water content (Brevik et al., 2006); thus, high soil EC_a values in this area were expected.

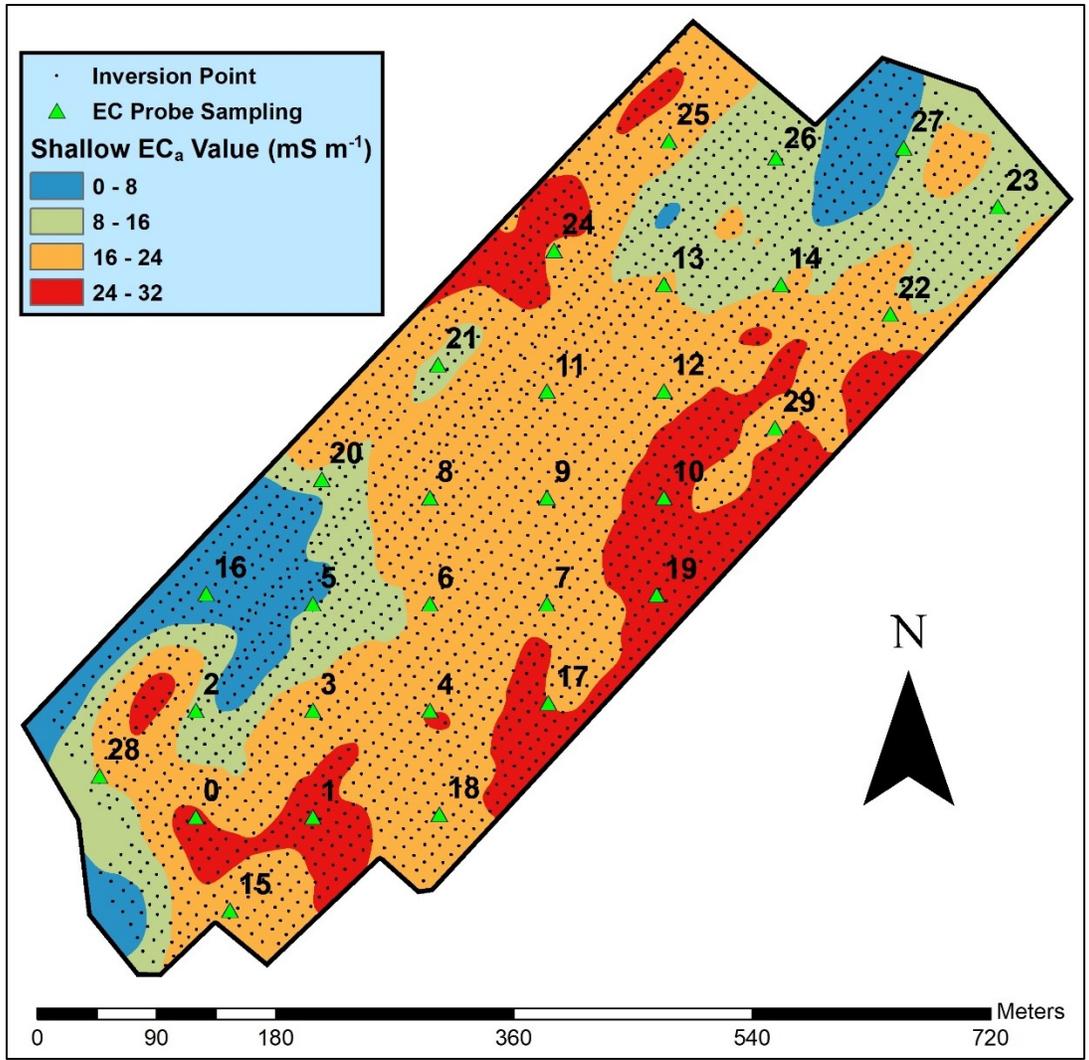


Figure 5-16. Shallow layer EC_a inversion map

The relatively low EC_a value in the south-west and north-east area of the field was consistent with the visual inspection of the field; in this region, the soil was relatively compact, sandy and had a gray-brownish color with little or no organic fibrous materials. The visual observation indicates that the soil was predominantly mineral. Therefore, it has low water holding capacity and consequently, results in low EC_a in this region. This indication was successfully shown in Figure 5-15, where in this region the depth of the muck soil layer was relatively shallow ranging from 0 to 30 cm.

The Brute-Force inversion result for deep EC_a is shown in Figure 5-17. In the center and south area of the field, the deep layer yields a lower EC_a value than the shallow layer. In this area, the low EC_a value was consistent with the previous findings where this region

has a thick muck layer of 120 to 150 cm (Figure 5-15) and high EC_a value of 18 to 33 $mS\ m^{-1}$ (Figure 5-16); in the shallow layer, this was due to the high water holding capacity from the high organic matter content. Therefore, it can be interpreted that in these areas, the two layers model was appropriate, and the deep soil layer was comprised of compacted subsoil or possibly, an igneous bedrock with low EC_a value (McNeill, 1980a).

The opposite layering where the deep layer yielded a higher EC_a value than the shallow layer was observed. On the south-west and north-east portions of the field, where the inversion algorithm results in a shallow muck soil depth of 0 to 30 cm (Figure 5-15) with a low EC_a value of 0 to 8 $mS\ m^{-1}$ (Figure 5-16), the deep EC_a value was higher ranging from 16 to 24 $mS\ m^{-1}$ (Figure 5-17). The low EC_a value in the shallow layer corresponds to the high soil mineral content, whereas the higher EC_a value resembles the high organic content in the deeper soil layer. The organic layer might be buried under a thin layer of mineral soil due to various processes, such as soil tillage, field preparation or other causes. Therefore, it might be possible to have three soil layers in this region comprised of mineral, organic and mineral/bedrock at the shallow, middle and deeper layer depths, respectively. A similar result was found in the north-east side of the field. However, the outcome in this area was probably related to the drainage trench and sprayed nursery field adjacent to this field. The sprayed water might have contributed to the higher soil EC_a values which were found in the deep layer. This idea of a dry organic soil layer on top of mineral subsoil with high water content or the opposite layering of high organic content buried under a thin mineral soil layer need further exploration. Different degrees of peat decomposition might alter the electrical conductivity of the soil profile (Walter et al., 2015; Walter et al., 2019). Therefore, in this scenario, it is impossible to determine the soil layering without referencing further geophysical sampling (e.g. deep pit excavation).

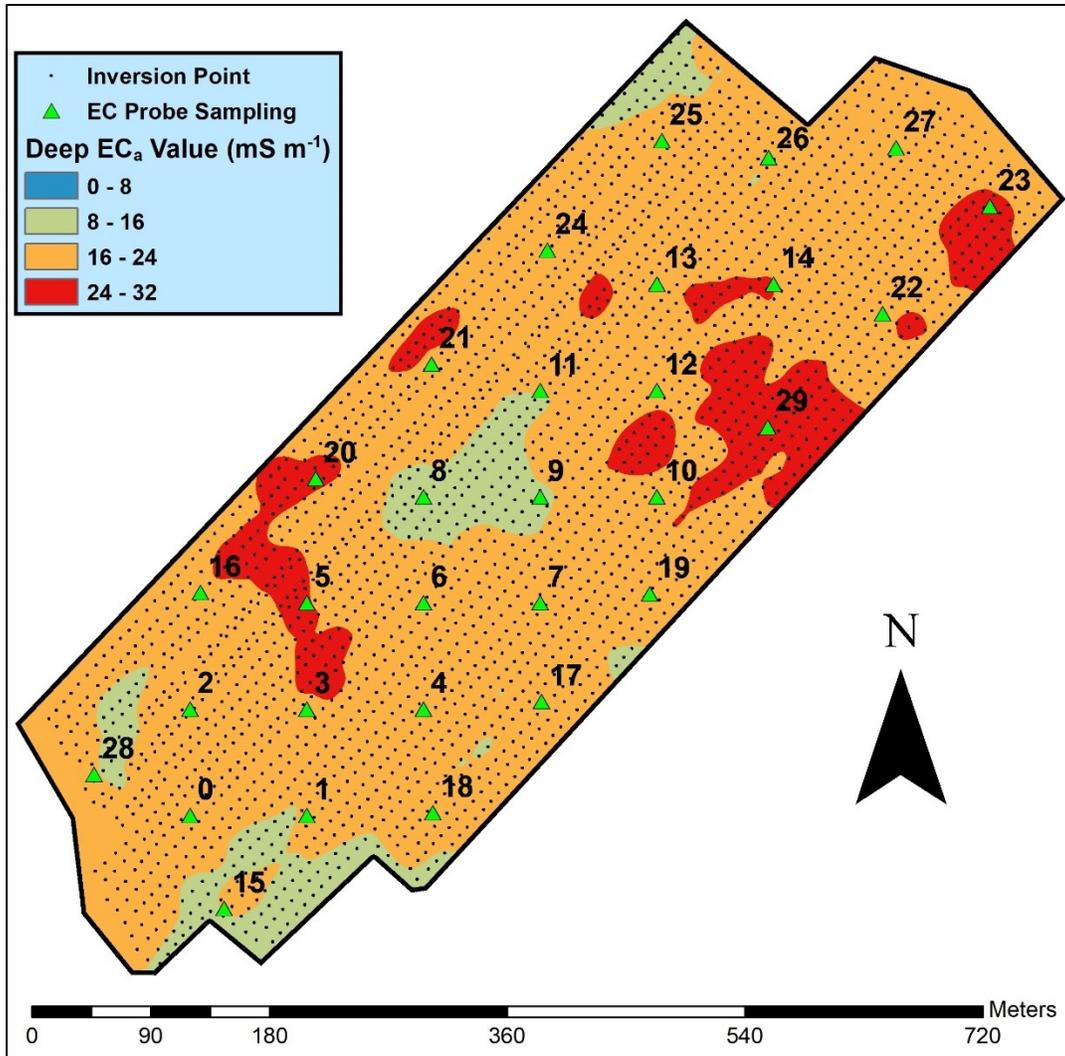


Figure 5-17. Deep EC_a inversion map

5.3.3. Soil EC_a Model to Determine the Depth of Muck Soil Layer

Figure 5-18 represents subsoil manual EC_a probe profiles and two-layer EC_a models together with Brute-Force inversion results in a location with deep muck soil layer depths. At these locations, the subsoil EC_a probe was able to penetrate deep into the soil profile. The EC_a model provides the appropriate two soil layer profile depth from the EC_a probe data. The EC_a layering models help to determine the boundary layer between the shallow muck and deeper soil layers. For example, Brute-Force inversion suggested that the depth of the shallow layer for sampling points 4, 17, 19 and 10 are 130, 130, 150 and 150 cm, respectively. From the EC_a modelling and profile information, the user can determine that the distinct EC_a profile change from the EC_a model was an indication of the boundary

layer between the muck and deeper soil layer. For these specific sampling locations, the appropriate depth of the muck layer was 70, 90, 100 and 150 cm, respectively.

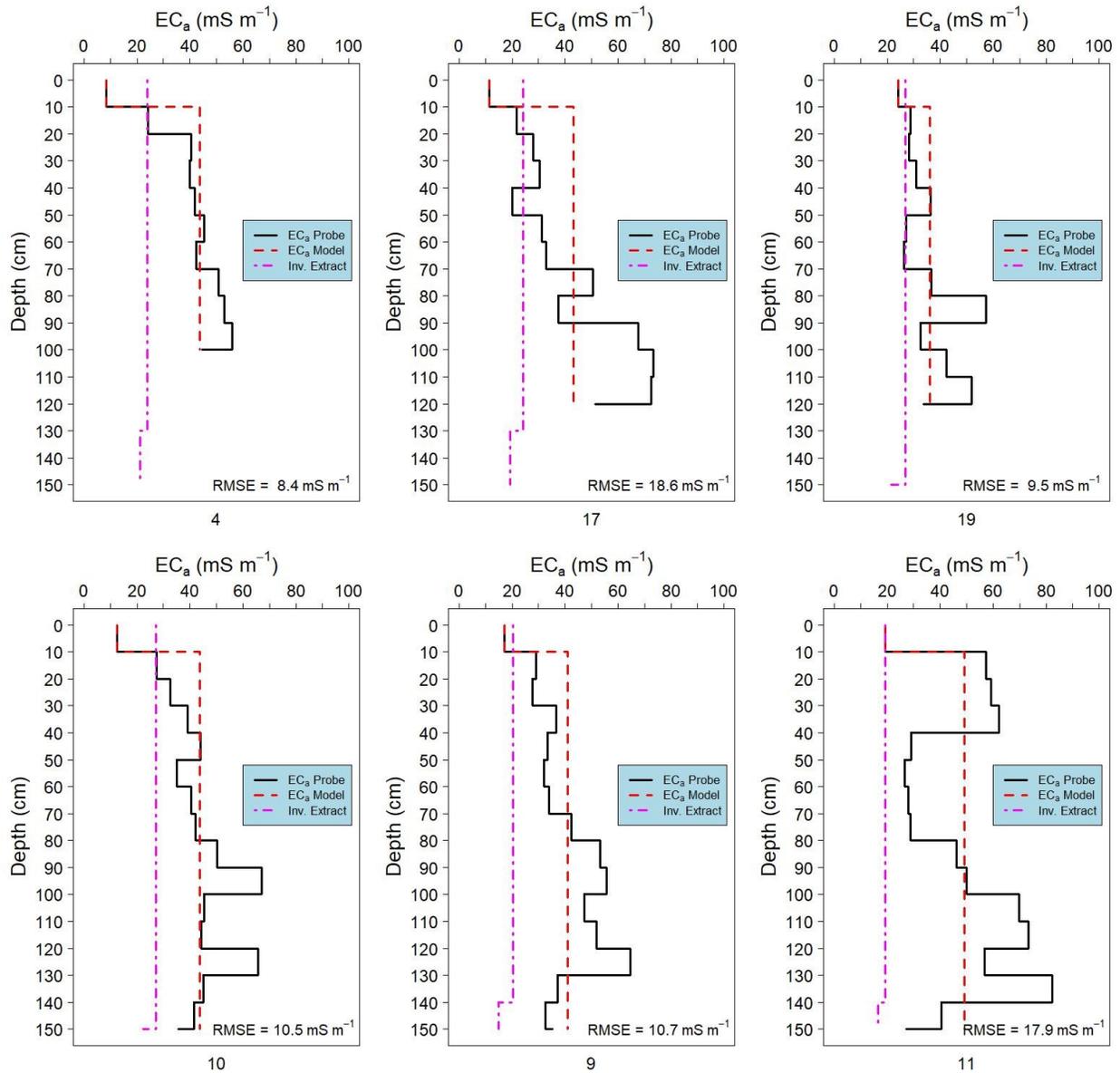


Figure 5-18. Soil EC_a profiles and models on locations with deep muck soil layer depth

Uniquely, for the sampling points 9 and 11, Brute-Force inversion resulted in the depth of the muck layer as 140 cm for both locations. However, the soil EC_a profiles consisted of three layers instead of two. Moreover, the EC_a profiles were opposite one another. At location 9, the EC_a profile was low-high-low whereas at location 11, the EC_a profile was high-low-high from shallow to deep soil depth. This finding indicates that there are

locations in the field with three soil layering. To perform inversion on a three soil layering scenario, additional soil EC_a sensors, such as the Veris 2000XA (Veris Technologies Inc., Salina, Kansas, USA) should be used together with the DUALEM-21S to provide more depth investigation (Sudduth et al., 2013). Alternatively, an EMI sensor with six depths of investigation such as the DUALEM-421S or the relatively new DUALEM-21HS (Dualem, Inc., Milton, Ontario, Canada) can also be used (Davies et al., 2015; Dabas et al., 2016). These sensor's configuration could potentially resolve the three soil layering cases and therefore, plan for the next stage of this research.

Figure 5-19 represents subsoil EC_a probe profiles and two-layer EC_a models together with Brute-Force inversion results at locations with shallow muck soil layer depths. Generally, the subsoil EC_a probe was able to penetrate the middle (60 to 120 cm) of the whole soil profile. From sampling points 20, 14, 29 and 23, the Brute-Force inversion resulted in the depth of the shallow layer as 20, 40, 50 and 20 cm, respectively. At these locations, the EC_a models suggested depths of 40, 70, 30 and 30 cm, respectively.

Interesting findings are shown at sampling locations 16 and 27 where the soil EC_a profile tended to become lower with depth. It is clearly shown in Figure 5-15 that at these locations, the depth of the muck layer was shallow (0 to 30 cm). In addition, the Brute-Force inversion result for shallow EC_a values at both locations (0 to 8 mS m⁻¹) were less than the deep EC_a values (16 to 24 mS m⁻¹), which indicates an inverted soil profile of mineral soil type above the muck soil layer. Alternatively, a three layers model discussed earlier might be appropriate in these locations. The suggested layering would be organic, a mix between mineral and organic, and mineral/bedrock at the shallow, middle, and deeper layer depths, respectively.

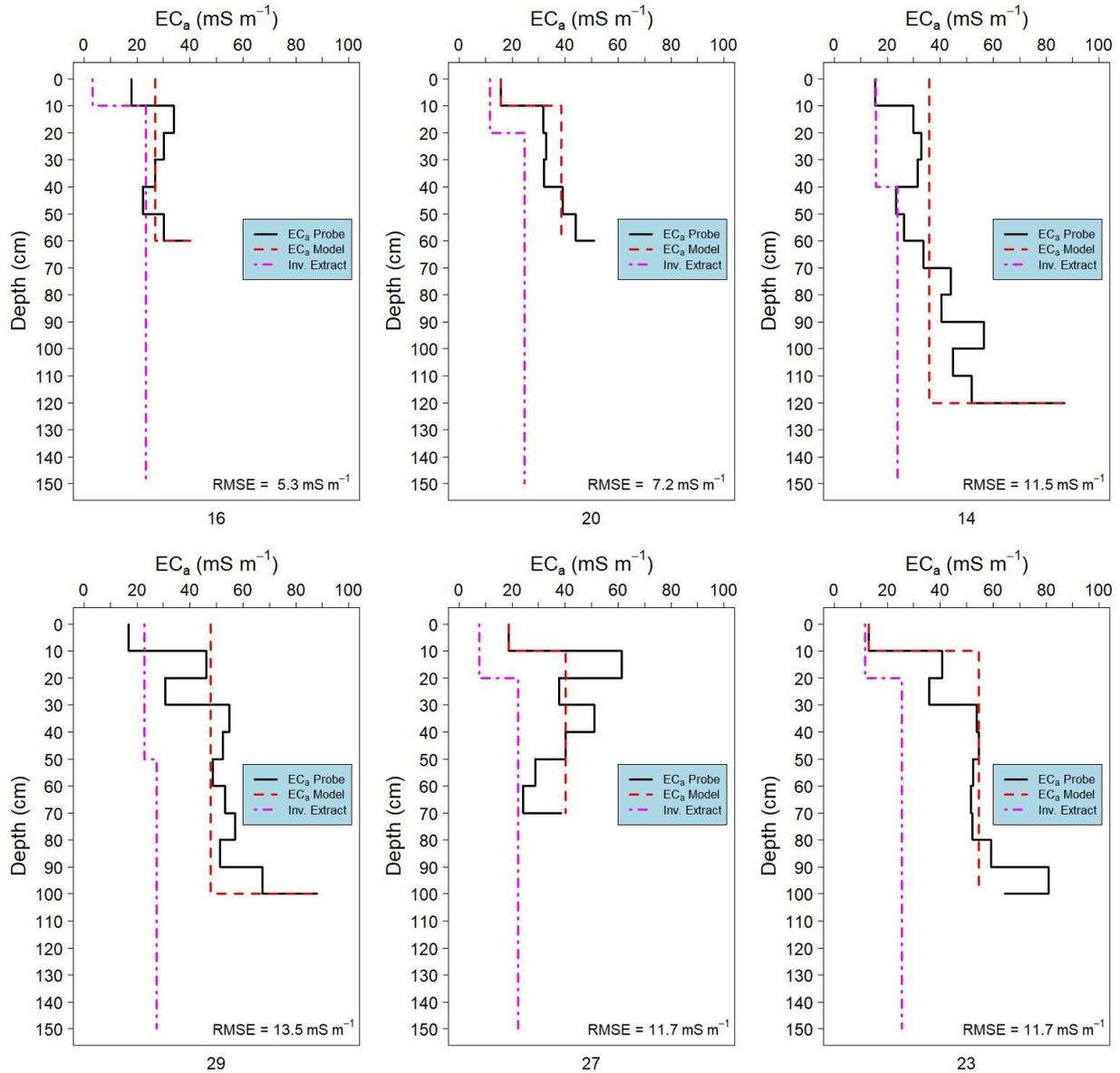


Figure 5-19. Soil EC_a profiles and models on locations with shallow muck soil layer depth

The EC_a probe was unable to penetrate deeper at locations 2, 21 and 5 (Figure 5-20). This might be a good indication that the soil mainly consisted of mineral soil which is prone to compaction and has low EC_a values, which was successfully shown in Figure 5-15 and Figure 5-16. Similarly, the manual EC_a probe could not penetrate the soil at locations 0, 1, 8, 24, 25 and 22. This is contrary to the Brute-Force inversion result (Figure 5-15) which suggests that at these locations, the depth of the muck soil was deep (90 to

150 cm) and therefore, should ease the EC_a probe penetration. A reversed soil EC_a profile layering might be suggested in these locations.

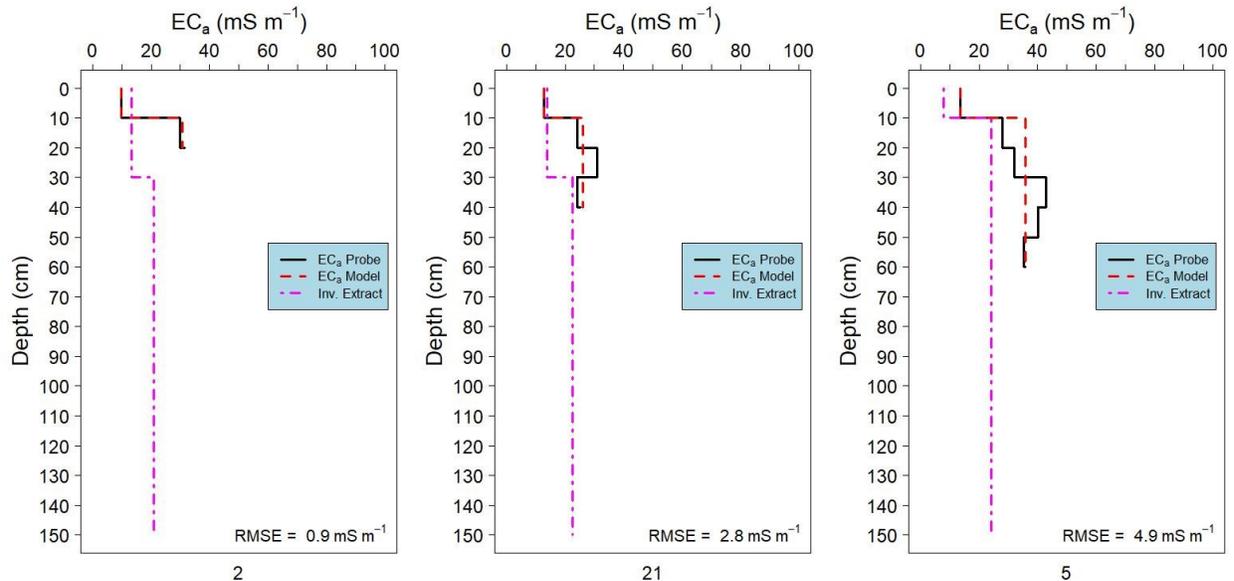


Figure 5-20. Soil EC_a profiles and models on locations with penetration restricting depth and shallow muck soil layer depth

From sampling points 3, 18, 6 and 7 (Figure 5-21), the Brute-Force inversion resulted in the depth of shallow layers of 80, 110, 100 and 100 cm, respectively. The EC_a models suggested depths of 80, 100, 100 and 90 cm, respectively. The Brute-Force inversion results were more consistent in determining the shallow and deep soil boundary layers. The EC_a models were able to help the user in determining the boundary layer depth at locations where the shallow and deep EC_a inversion values were undistinguishable such as at location 7. Again, from locations 15 and 12, three soil EC_a layers occurred with low, high and low EC_a at shallow, middle and deep soil layers.

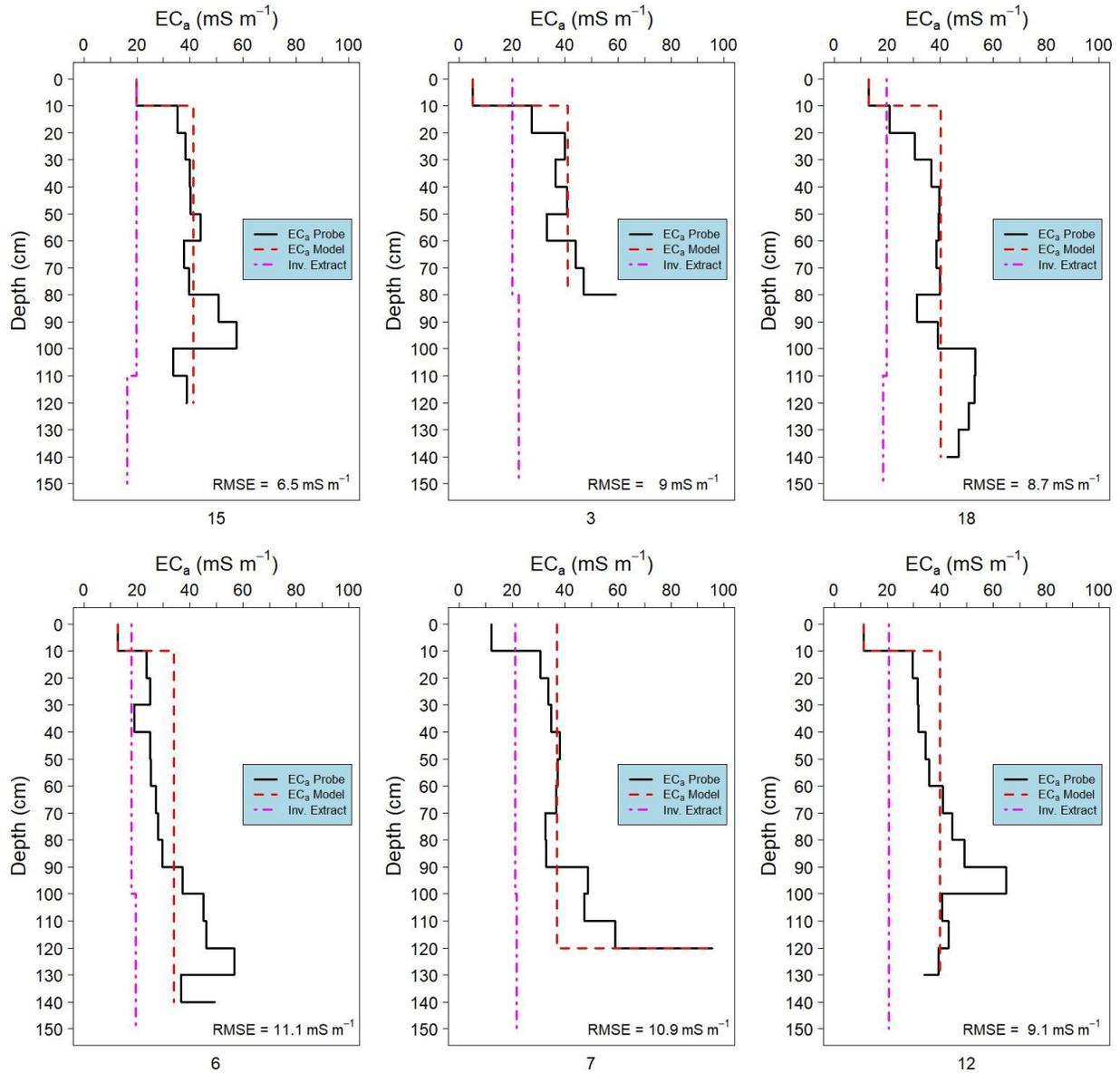


Figure 5-21. EC_a profiles and models on locations with moderate topsoil layer depth

5.3.4. Depth of Muck Soil Layer Estimation

The depth of the muck soil layer from EC_a models and the corresponding Brute-Force inversion results are summarized in Table 5-1. At locations with an undefined depth of muck soil layer due to penetration restricting depth, three layers or reversed EC_a profiles trend were removed. The depth of the muck soil layer estimation from EC_a models and Brute-Force inversion were compared and shown in Figure 5-22. From this figure, the Brute-Force inversion was able to determine the soil boundary depth of the two EC_a layers

profile with decent results (R^2 of 0.76). The Brute-Force inversion slightly underestimated the depth of the muck layer at 80 to 120 cm depth. This suggests that there are more soil processes that contribute to soil EC_a variations, such as water percolation, localized compaction and mineral content at this soil layer. Additionally, soil excavation or soil spectra profiling should be conducted to validate the Brute-Force inversion depth of the muck soil layer result.

Table 5-1. Depth of muck soil estimation from EC_a model and Brute-Force inversion

Location	Muck Soil Depth (cm)		Remarks
	EC_a Model	Inversion	
0	undefined	150	Penetration restricting depth
1	undefined	150	Penetration restricting depth
2	20	30	
3	80	80	
4	70	130	
5	30	10	
6	100	100	
7	90	100	
8	undefined	150	Penetration restricting depth
9	undefined	140	Three layers profile
10	150	150	
11	undefined	140	Three layers profile
12	undefined	90	Reversed EC_a trend
13	undefined	70	Penetration restricting depth
14	70	40	
15	undefined	110	Reversed EC_a trend
16	undefined	10	Reversed EC_a trend
17	90	130	
18	100	110	
19	100	150	
20	40	20	
21	20	30	
22	undefined	90	Penetration restricting depth
23	30	20	
24	undefined	90	Penetration restricting depth
25	undefined	130	Penetration restricting depth
26	undefined	90	Penetration restricting depth
27	undefined	20	Reversed EC_a trend
28	undefined	100	Penetration restricting depth
29	30	50	

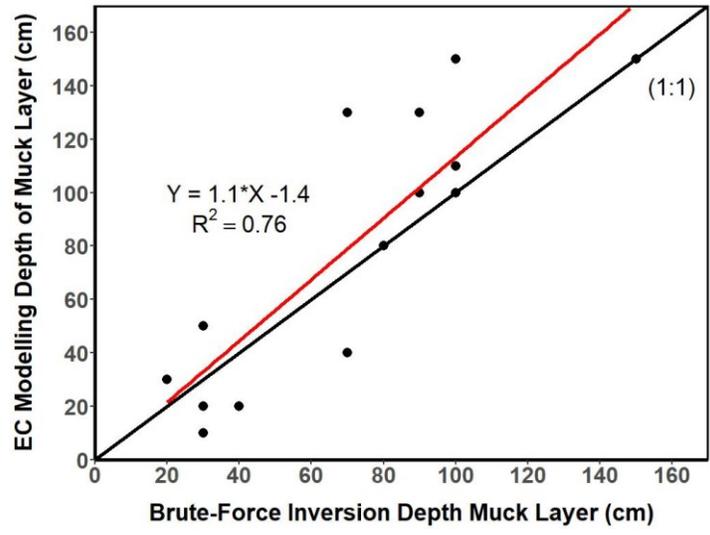


Figure 5-22. Comparison of muck soil layer depth estimation between Brute-Force inversion and EC_a modelling

In locations where the EC_a probe could not penetrate the entire soil profile, an effort to shift the sampling location to a new location within a 50 cm radius of the original location was tried but not all of these locations resulted in better penetration. Since the time needed to perform the full soil EC_a profiling was quite long, redoing the sampling with the current EC_a probe design might not be effective and thus, the idea of mechanically operated deep EC_a probe might be more relevant. For validation purposes, an alternative geophysical method, such as deep pit/trench excavation or direct push soil sampling (ASTM, 2014), is suggested to determine the soil layering depths and horizon designation (Watson, 2007). Unfortunately, in this study, that information was not known and therefore, the geophysical properties were unknown.

5.4. Conclusion

The new DUALEM-21S filtering software was designed to improve the quality of conventional EC_a data valuable for both inversion and other analysis, such as subsoil feature detection and soil properties modelling. This software also provides a fast, robust and less laborious effort to filter the high density DUALEM-21S data. Through another software developed in this project, the Brute-Force EC_a inversion was successfully implemented to determine the overall structure of a simplified soil profile that can be used for directed sampling of soil for agronomic recommendations. As shown in the example case study, this software can also assist farmers in characterizing the organic matter status of their field. The GUI of both software packages was built to be intuitive and easy to operate. The Brute-Force EC_a inversion results are discrete with a distinct depth and EC_a transition. The inversion algorithm was successful in depicting the muck soil status in the field. The soil EC_a modelling from the specially designed subsoil manual EC_a probe data was able to provide guidance in determining the appropriate soil transition layer between the shallow muck soil and the deeper clay soil layer. The depth of the muck soil layer was in accordance with the value found in the shallow and deep soil EC_a inversion result.

6. Summary and General Conclusion

6.1. Summary

This research was conducted to develop soil sensing platforms to perform direct soil analysis of selected soil properties for use in different field settings. An on-the-spot soil analyzer was developed to accommodate large farms with vast areas while a manual soil analyzer was tailored to provide portability in soil chemical scouting. Additionally, to facilitate direct soil measurements (DSM) of chemical soil attributes, a soil EC_a data processing and interpretation (e.g. inversion) algorithm was developed.

In the first study, an automated, on-the-spot soil analyzer (OSA) was developed and tested. The OSA prepared the soil test surface at a predefined depth and deployed multiple sensors to perform measurements in real time. The optimization of the OSA design resulted in a system that uses parallel linkages to reduce digging curvature and includes an additional linear actuator to automatically cover the sampling hole. Several alternative controller algorithms were attempted while selecting a robust set of OSA operation parameters for various field conditions. Typical digging time was less than 15 s and the entire OSA operation (digging, measurements, washing and covering the sampling hole) took under 60 s to complete, as was specified in the design requirements. To illustrate OSA performance, pH and nitrate ion-selective electrodes (ISEs) were tested in fields with ongoing soil fertility trials. When comparing OSA measurements with reference laboratory tests, the coefficient of determination for a simple linear regression was 0.59 and 0.72, respectively with RMSE of 0.43 pH and 0.16 pNO₃. The Arduino based data acquisition system (DAQ) was designed to support up to four ISEs measurements simultaneously with the capability to integrate additional sensors, such as EC_a, moisture/temperature, optical reflectance, etc. All the OSA procedures and ISEs responses were controlled and monitored via a developed graphical user interface (GUI) from the operator's compartment. The OSA GUI provided an automatic one-button operation for the operator to complete all soil analyzing procedures. The OSA was equipped with a camera to monitor the digging process and, if needed, to alter the parameters of the automatic measurement cycle in real time. From the field testing, it was

determined that the camera and manual adjustment of the operation parameters were valuable in stony or dry field conditions.

In the second study, an integrated system to perform simultaneous measurements of multiple soil chemical properties was designed based on a handheld soil sampler. As an alternative to the OSA, the portable soil analyzer offers rapid quantification of spatial soil chemical variability which is valuable for providing supportive information for site-specific management in small plots and in specialty cropping system scenarios. The manual soil analyzer DAQ is able to perform four simultaneous soil chemical analysis. The DAQ provides a noise, ground loop and crosstalk protection which are essential for direct soil measurement using multiple ISEs. In the field testing, the soil sensing platform was convenient to use and provided good ISEs contact as well as reliable data acquisition. Together with the multi-channel DAQ, the manual soil analyzer can be used for chemical profiling at a maximum depth of 25 cm by setting all four of the DAQ's channels to use a similar ISE. Similar to the OSA, the field test of the manual soil analyzer yielded relationships between field and lab soil pH and nitrate ion activity with a R^2 between 0.53 to 0.88 with RMSE ranging from 0.17 to 0.36 pH and R^2 between 0.83 and 0.84 with a RMSE of 0.21 to 0.29 pNO_3 for soil nitrate. This result was comparable to the existing benchtop soil analyzers. In addition, the manual soil analyzer offers simpler soil analysis procedures compared to the current portable soil chemical analyzer. Instead of conducting soil sampling and analyzing using separate instruments, the manual soil analyzer developed in this thesis can perform those tasks simultaneously.

In the third study, to support the soil sensing platform's operation, a Brute-Force apparent soil electrical conductivity (EC_a) inversion was developed. The high density DUALEM-21S (Dualem, Inc. Milton, Ontario, Canada) EC_a data were filtered to ensure the valid data is in the low induction number (LIN) range. The EC_a inversion algorithm does not require initial EC_a and depth models. Instead, it relies on an exhaustive method to explore the defined soil EC_a model layering. Since the DUALEM-21S has four simultaneous EC_a measurements corresponding to different depths of investigations, a two-layer soil EC_a model was selected to represent a shallow (down to 150 cm depth) and deeper subsoil. The Brute-Force EC_a inversion depicts the depth of each layer and

its corresponding EC_a value. The algorithm will start to search for the best shallow and deep layer depth combination based on the RMSE minimization between the modelled two-layer EC_a values and DUALEM-21S measurements. The depth combination was increased by 10 cm until it reached 150 cm from the surface and the modelled two-layer EC_a value was set to 200 mS m^{-1} maximum with 0.2 mS m^{-1} increments. The inverted DUALEM-21S was discreet and needed to be geostatistically interpolated to make a realistic representation of the field. The Brute-Force EC_a inversion was successfully tested to estimate the volume of the shallow muck soil layer and to provide realistic maps of shallow and deep layer EC_a values. These maps provide sampling location suggestions for the farmers to further assess different soil chemical properties using the developed soil sensing platforms and offers convenience in interpreting soil EC_a measurements.

6.2. General Conclusion

Soil sensing platforms developed in this thesis are an important research phase for producing high quality soil properties data. In the current settings, the platforms use the ISE sensor for assessing soil chemical properties directly in the field. The OSA platform was able to accommodate additional proximal soil sensors for improved soil properties data collection and estimation. The current mobile soil sensing platforms' handicap of low sampling quantity and soil nutrient data quality have been addressed through a direct soil measurement methodology used in the OSA and the manual soil analyzer. Both platforms offer options for conducting soil chemical measurements on different field scales and under varied conditions. The Brute-Force soil EC_a inversion algorithm provides a support system for the soil sensing platform's operation by providing soil EC_a data processing tools for easier EC_a data interpretation and for determining appropriate sampling locations and soil attributes pertaining to soil chemical properties data collected by the soil sensing platforms. The comprehensive precision agriculture methodologies developed in this thesis successfully collect and deliver important soil properties information valuable for further agronomical decision strategies. In addition, as the methodologies developed in this thesis are related to the determination of important soil attributes, they provide the versatility to be employed for other purposes, such as environmental assessment (e.g. land reclamation, forest organic layer estimation, waste leaching potential, etc.)

7. Contributions to Knowledge and Suggestions for Future Research

7.1. Contribution to Knowledge

The thesis contributes mainly to the development of coherence methodologies to characterize soil chemical properties directly in the field. The aspiration for sustainable farming was achieved with the development of robust soil sensing platforms for direct soil chemical measurements. In general, the thesis offers robust proximal soil sensing platforms and supporting algorithms intended to provide high quality soil chemical data and related soil properties analysis for various farm management activities. The specific contributions to the knowledge from this thesis are the following:

1. Developed a dependable soil sensing platform for georeferenced (on-the-spot) variable depth measurements of multiple soil properties directly in the field. The new OSA design offers a multiple proximal sensors platform, interoperability and operational convenience which are essential for farmers and, therefore, has the potential for commercialization. The developed OSA controller provides an optimized soil measurement cycle for any typical soil surface with minimal soil distortion.
2. Developed a unique portable soil sensing platform suitable for streamlined soil sampling and chemical analysis procedures. The manual soil analyzer offers simultaneous multiple depth soil sampling and chemical analysis using a combination of commercial ISEs.
3. Developed a low-cost and compact Arduino-based data acquisition system for direct soil measurement using multiple ISEs used to obtain georeferenced measurements of selected ions activity in challenging environments, such as field conditions.
4. Developed a free and easy to use integrated data processing software for filtering and inverting the on-the-go soil EC_a data collected using the popular DUALEM-21S electromagnetic induction (EMI) sensor. The Brute-Force soil EC_a inversion algorithm processes the raw EC_a data into the modelled two-layer EC_a maps with additional depth of shallow layer information which represents important

parameters of the soil profile valuable for determining soil properties relevant to further agronomical practices.

5. Developed a unique portable soil EC_a sensor for direct soil EC_a measurements at different depths down to 1.5 m and demonstrated that the instrument can be used to validate the maps generated through soil EC_a data processed using the Brute-Force soil EC_a inversion algorithm developed in this thesis.

7.2. Suggestions for Future Research

Future research for both soil sensing platforms should focus on collecting more field evaluation data in different field conditions. Integrating new and better proximal soil sensors to the OSA is essential for a comprehensive soil property analysis. The following are detailed suggestions for future research identified from this thesis:

1. Integrate additional proximal soil sensors suitable for measuring specific soil properties using OSA (e.g. optical spectroscopy for organic matter and texture, contact EC_a sensor for salinity and CEC, gamma-ray sensor for water content determination, etc.). This fusion of sensors is essential to create better and more comprehensive prediction of soil attributes to support various farm management decisions (e.g. determining the optimum rate of lime and fertilizer).
2. Perform further data integration and processing using information gathered from the OSA and manual soil analyzer, soil EC_a inversion and other precision agriculture data (e.g. plant canopy indices, crop yield and other soil properties data) to optimize the soil sampling quantity as well as the calibration of samples distribution.
3. With the advances in space exploration technology and artificial intelligence, the development of an OSA robot platform capable of conducting a fully autonomous direct soil sensing operation is encouraged. Additionally, a more reliable, higher sampling rate, low noise floor current sensor is required to provide faster OSA digging controller response. An alternative digging depth sensing mechanism or sensors need to be explored to provide reliable digging depth feedback, especially in uneven terrain conditions.

4. A three layers soil EC_a model and appropriate EMI sensors (e.g. DUALEM-421S) need to be developed and used for improved Brute-Force EC_a inversion capability. Multiple depth soil spectral measurements or excavations are advised to validate the EC_a layering of the Brute-Force EC_a inversion results.

8. References

- Adamchuk, V., Dhawale, N., & Rene-Laforest, F. Development of on-the-spot soil analyzer for measuring soil chemical properties. In *12th International Conference on Precision Agriculture, Sacramento, California, USA, 20-23 July 2014*: International Society of Precision Agriculture
- Adamchuk, V. I. (2005). Portable probe for on-the-spot measurement of soil pH. (pp. 1-10). Lincoln, Nebraska, USA: University of Nebraska–Lincoln.
- Adamchuk, V. I., Hummel, J. W., Morgan, M. T., & Upadhyaya, S. K. (2004). On-the-go soil sensors for precision agriculture. *Computers and Electronics in Agriculture*, *44*(1), 71-91, doi:10.1016/j.compag.2004.03.002.
- Adamchuk, V. I., Lund, E. D., Reed, T. M., & Ferguson, R. B. (2007). Evaluation of an on-the-go technology for soil pH mapping. *Precision Agriculture*, *8*(3), 139-149, doi:10.1007/s11119-007-9034-0.
- Adamchuk, V. I., Lund, E. D., Sethuramasamyraja, B., Morgan, M. T., Dobermann, A., & Marx, D. B. (2005). Direct measurement of soil chemical properties on-the-go using ion-selective electrodes. *Computers and Electronics in Agriculture*, *48*(3), 272-294, doi:10.1016/j.compag.2005.05.001.
- Adamchuk, V. I., Morgan, M. T., & Ess, D. R. (1999). An Automated Sampling System For Measuring Soil pH. *Transaction of the ASAE*, *42*(4), doi:10.13031/2013.13268.
- Adamchuk, V. I., & Viscarra Rossel, R. A. (2010). Development of On-the-Go Proximal Soil Sensor Systems. In R. A. V. Rossel, A. B. McBratney, & B. Minasny (Eds.), *Proximal Soil Sensing* (pp. 15-28). Dordrecht: Springer Netherlands.
- Addiscott, T. (2005). *Nitrate, Agriculture and the Environment*. Wallingford, GB: CAB International.
- Adsett, J. F., Thottan, J. A., & Sibley, K. J. (1999). Development of an Automated On-the-Go Soil Nitrate Monitoring System. *Applied Engineering in Agriculture*, *15*(4), doi:10.13031/2013.5789.
- Allred, B. J., Daniels, J. J., & Ehsani, M. R. (2008). Resistivity and Electromagnetic Induction Case Histories. In *Handbook of Agricultural Geophysics* (Books in Soils, Plants, and the Environment): CRC Press.
- Angelopoulou, T., Balafoutis, A., Zalidis, G., & Bochtis, D. (2020). From Laboratory to Proximal Sensing Spectroscopy for Soil Organic Carbon Estimation—A Review. *Sustainability*, *12*(2), 443.
- Artigas, J., Beltran, A., Jiménez, C., Baldi, A., Mas, R., Domínguez, C., et al. (2001). Application of ion sensitive field effect transistor based sensors to soil analysis. *Computers and Electronics in Agriculture*, *31*(3), 281-293, doi:10.1016/S0168-1699(00)00187-3.
- ASTM (2014). Standard Guide for Direct Push Soil Sampling for Environmental Site Characterizations. (Vol. D6282 / D6282M-14). West Conshohocken, PA: ASTM International.
- Bakker, E. (2019). Ion-Selective Electrodes | Overview. In P. Worsfold, C. Poole, A. Townshend, & M. Miró (Eds.), *Encyclopedia of Analytical Science (Third Edition)* (pp. 231-251). Oxford: Academic Press.
- Baxter, S. J., Oliver, M. A., & Gaunt, J. (2003). A Geostatistical Analysis of the Spatial Variation of Soil Mineral Nitrogen and Potentially Available Nitrogen Within an Arable Field. *Precision Agriculture*, *4*(2), 213-226, doi:10.1023/a:1024565507688.
- Birrell, S. J., & Hummel, J. W. (2001). Real-time multi ISFET/FIA soil analysis system with automatic sample extraction. *Computers and Electronics in Agriculture*, *32*(1), 45-67, doi:10.1016/S0168-1699(01)00159-4.
- Bobryk, C. W., Yost, M. A., & Kitchen, N. R. (2018). Field variability and vulnerability index to identify regional precision agriculture opportunity. *Precision Agriculture*, *19*(4), 589-605, doi:10.1007/s11119-017-9541-6.

- Bouwman, A. F., Beusen, A. H. W., Lassaletta, L., Apeldoorn, D. F. v., Grinsven, H. J. M. v., Zhang, J., et al. (2017). Lessons from temporal and spatial patterns in global use of N and P fertilizer on cropland.
- Brevik, E. C., Fenton, T. E., & Lazari, A. (2006). Soil electrical conductivity as a function of soil water content and implications for soil mapping. *Precision Agriculture*, 7(6), 393-404, doi:10.1007/s11119-006-9021-x.
- Bronson, K. F., Booker, J. D., Officer, S. J., Lascano, R. J., Maas, S. J., Searcy, S. W., et al. (2005). Apparent Electrical Conductivity, Soil Properties and Spatial Covariance in the U.S. Southern High Plains. *Precision Agriculture*, 6(3), 297-311, doi:10.1007/s11119-005-1388-6.
- Brouder, S. M., Thom, M., Adamchuck, V. I., & Morgan, M. T. (2003). Potential Uses of Ion-Selective Potassium Electrodes in Soil Fertility Management. *Communications in Soil Science and Plant Analysis*, 34(19-20), 2699-2726, doi:10.1081/CSS-120025214.
- Bunting, L., Leavitt, P. R., & Wissel, B. (2011). Sudden Ecosystem State Change in Lake Winnipeg, Canada, Caused by Eutrophication Arising from Crop and Livestock Production during the 20th Century. Winnipeg, CA: Government of Manitoba.
- Callegary, J., Ferré, T., & Groom, R. (2007). Vertical Spatial Sensitivity and Exploration Depth of Low-Induction-Number Electromagnetic-Induction Instruments. *Vadose Zone*, 6, 158-167, doi:10.2136/vzj2006.0120.
- Cardwell, T. J., Cattrall, R. W., Hauser, P. C., & Hamilton, I. C. (1988). A multi-ion sensor cell and data-acquisition system for flow injection analysis. *Analytica Chimica Acta*, 214, 359-366, doi:10.1016/S0003-2670(00)80456-0.
- Christiansen, A., Pedersen, J., Auken, E., Sørensen, N., Holst, M., & Kristiansen, S. (2016). Improved Geoarchaeological Mapping with Electromagnetic Induction Instruments from Dedicated Processing and Inversion. *Remote Sensing*, 8, 1022, doi:10.3390/rs8121022.
- Christy, C. D. (2008). Real-time measurement of soil attributes using on-the-go near infrared reflectance spectroscopy. *Computers and Electronics in Agriculture*, 61(1), 10-19, doi:10.1016/j.compag.2007.02.010.
- Clearwater, R. L., Martin, T., & Hoppe, T. (2016). Environmental sustainability of Canadian agriculture: Agri-environmental indicator report series – Report #4. Ottawa, ON: Agriculture and Agri-Food Canada.
- Conkling, B. L., & Blanchar, R. W. (1988). A Comparison of pH Measurements using the Antimony Microelectrode and Glass Electrode. *Agronomy*, 80(2), 275-278, doi:10.2134/agronj1988.00021962008000020025x.
- Conkling, B. L., & Blanchar, R. W. (1989). Glass Microelectrode Techniques for In Situ pH Measurements. *Soil Science Society of America*, 53(1), 58-62, doi:10.2136/sssaj1989.03615995005300010011x.
- Constable, S. C., Parker, R. L., & Constable, C. G. (1987). Occam's inversion: A practical algorithm for generating smooth models from electromagnetic sounding data. *Geophysics*, 52(3), 289-300.
- Corwin, D. L. (2008). Past, Present, and Future Trends of Soil Electrical Conductivity Measurement Using Geophysical Methods. In B. J. Allred, J. J. Daniels, & M. R. Ehsani (Eds.), *Handbook of Agricultural Geophysics* (pp. 17-44, Books in Soils, Plants, and the Environment): CRC Press.
- Corwin, D. L., & Lesch, S. M. (2003). Application of Soil Electrical Conductivity to Precision Agriculture. *Agronomy*, 95(3), 455-471, doi:10.2134/agronj2003.4550.
- Corwin, D. L., Lesch, S. M., Shouse, P. J., Soppe, R., & Ayars, J. E. (2008). Delineating Site-Specific Management Units Using Geospatial ECa Measurements. In B. J. Allred, J. J. Daniels, & M. R. Ehsani (Eds.), *Handbook of Agricultural Geophysics* (pp. 247-254, Books in Soils, Plants, and the Environment): CRC Press.

- Dabas, M., Anest, A., Thiesson, J., & Tabbagh, A. (2016). Slingram EMI Devices for Characterizing Resistive Features Using Apparent Conductivity Measurements: check of the DualEM-421S Instrument and Field Tests. *Archaeological Prospection*, 23(3), 165-180.
- Daniels, J., Vendl, M., Reza Ehsani, M., & Allred, B. (2008). Electromagnetic Induction Methods. In B. J. Allred, J. J. Daniels, & M. R. Ehsani (Eds.), *Handbook of Agricultural Geophysics* (pp. 109-128, Books in Soils, Plants, and the Environment): CRC Press.
- Davenport, J. R., & Jabro, J. D. (2001). Assessment of hand held ion selective electrode technology for direct measurement of soil chemical properties. *Communications in Soil Science and Plant Analysis*, 32(19-20), 3077-3085, doi:10.1081/CSS-120001108.
- Davies, G., Huang, J., Monteiro Santos, F. A., & Triantafyllis, J. (2015). Modeling Coastal Salinity in Quasi 2D and 3D Using a DUALEM-421 and Inversion Software. *Groundwater*, 53(3), 424-431, doi:10.1111/gwat.12231.
- De Smedt, P., Saey, T., Lehouck, A., Stichelbaut, B., Meerschman, E., Islam, M. M., et al. (2013). Exploring the potential of multi-receiver EMI survey for geoarchaeological prospection: A 90 ha dataset. *Geoderma*, 199, 30-36, doi:10.1016/j.geoderma.2012.07.019.
- Dhawale, N. (2015). Advances in proximal soil sensing through integrated systems approach.
- Dillingham, P. W., Radu, T., Diamond, D., Radu, A., & McGraw, C. M. (2012). Bayesian Methods for Ion Selective Electrodes. *Electroanalysis*, 24(2), 316-324, doi:10.1002/elan.201100510.
- Dimitrakopoulos, L. T., & Dimitrakopoulos, T. (2001). Evaluation of a Four Sensor Array Used in a Wall-Jet Configured Flow Cell for Flow Injection Potentiometry. *Electroanalysis*, 13(2), 161-163.
- Ding, X., Wei, C., Wang, R., Liao, X., & Li, S. (2014). Phosphorus Leaching Risk Assessment with Manure Fertilizer Application in South China. *Bulletin of Environmental Contamination and Toxicology*, 93(1), 120-125, doi:10.1007/s00128-014-1262-1.
- Domingue, K. J., Price, R. R., & Mailander, M. P. (2005). *Real-time Soil Nitrate Sensing*. St. Joseph, MI, Doolittle, J. A., & Brevik, E. C. (2014). The use of electromagnetic induction techniques in soils studies. *Geoderma*, 223-225, 33-45, doi:10.1016/j.geoderma.2014.01.027.
- DUALEM (2014). DUALEM-42 User's Manual. Canada: Dualem Inc.
- DUALEM (2018). Electromagnetic Induction: Operation and Interpretation especially at Low Induction Numbers. <http://www.dualem.com/eioi.htm>. Accessed 18 July 2018.
- Ehsani, M. R., Upadhyaya, S. K., Slaughter, D., Shafii, S., & Pelletier, M. (1999). A NIR Technique for Rapid Determination of Soil Mineral Nitrogen. *Precision Agriculture*, 1(2), 219-236, doi:10.1023/A:1009916108990.
- EMTOMO (2012). <http://www.emtomo.com/home/>. Accessed 15 April 2017.
- Ferguson, R. B. (2019). Proximal crop sensing. In J. Stafford (Ed.), *Precision agriculture for sustainability* (Vol. 1 online resource (xix, 494 pages) : color illustrations, Burleigh Dodds series in agricultural science ; number 52). Philadelphia, PA 19102-3406, USA: Burleigh Dodds Science Publishing.
- Fleming, K. L., Westfall, D. G., Wiens, D. W., & Brodahl, M. C. (2000). Evaluating Farmer Defined Management Zone Maps for Variable Rate Fertilizer Application. *Precision Agriculture*, 2(2), 201-215, doi:10.1023/a:1011481832064.
- Fridgen, J. J., Kitchen, N. R., Sudduth, K. A., Drummond, S. T., Wiebold, W. J., & Fraisse, C. W. (2004). Management zone analyst (MZA). *Agronomy*, 96(1), 100-108.
- Fuglie, K. O. (2010). Sources of growth in Indonesian agriculture. *Journal of Productivity Analysis*, 33(3), 225-240, doi:10.1007/s11123-009-0150-x.
- Ge, Y., Thomasson, J. A., & Sui, R. (2011). Remote sensing of soil properties in precision agriculture: A review. *Frontiers of Earth Science*, 5(3), 229-238, doi:10.1007/s11707-011-0175-0.
- Hanks, R. J. (1992). Soil Heat Flow and Temperature. In *Applied Soil Physics: Soil Water and Temperature Applications* (pp. 139-159). New York, NY: Springer New York.

- Hansen, L. B., & Hansen, L. G. (2014). Can Non-point Phosphorus Emissions from Agriculture be Regulated Efficiently Using Input-Output Taxes? *Environmental and Resource Economics*, 58(1), 109-125, doi:10.1007/s10640-013-9693-4.
- Harmon, R. S., De Lucia, F. C., Miziolek, A. W., McNesby, K. L., Walters, R. A., & French, P. D. (2005). Laser-induced breakdown spectroscopy (LIBS) – an emerging field-portable sensor technology for real-time, in-situ geochemical and environmental analysis. *Geochemistry: Exploration, Environment, Analysis*, 5(1), 21-28, doi:10.1144/1467-7873/03-059.
- Havlin, J. L., Tisdale, S. L., Nelson, W. L., & Beaton, J. D. (2016). *Soil fertility and fertilizers*: Pearson Education India.
- Heil, K., & Schmidhalter, U. (2017). The Application of EM38: Determination of Soil Parameters, Selection of Soil Sampling Points and Use in Agriculture and Archaeology. *Sensors*, 17(11), 2540, doi:10.3390/s17112540.
- Huang, J., Scudiero, E., Choo, H., Corwin, D. L., & Triantafyllis, J. (2016). Mapping soil moisture across an irrigated field using electromagnetic conductivity imaging. *Agricultural Water Management*, 163, 285-294, doi:10.1016/j.agwat.2015.09.003.
- Hudson, B. D. (1994). Soil organic matter and available water capacity. *Journal of Soil and Water Conservation*, 49(2), 189-194.
- IPNI, I. P. N. I. (2012). *4R Plant Nutrition : a manual for improving the management of plant nutrition North American Version*. Norcross, GA: International Plant Nutrition Institute.
- Jahn, B. R., Linker, R., Upadhyaya, S. K., Shaviv, A., Slaughter, D. C., & Shmulevich, I. (2006). Mid-infrared Spectroscopic Determination of Soil Nitrate Content. *Biosystems Engineering*, 94(4), 505-515, doi:10.1016/j.biosystemseng.2006.05.011.
- Jelsma, I., Woittiez, L. S., Ollivier, J., & Dharmawan, A. H. (2019). Do wealthy farmers implement better agricultural practices? An assessment of implementation of Good Agricultural Practices among different types of independent oil palm smallholders in Riau, Indonesia. *Agricultural Systems*, 170, 63-76, doi:10.1016/j.agsy.2018.11.004.
- Jianhan, L., Maohua, W., Mao, Z., & Yane, Z. (2007). Development and modelling of a soil nitrate and potassium simultaneous rapid detection system based on ion selective electrodes. *New Zealand Journal of Agricultural Research*, 50(5), 635-640, doi:10.1080/00288230709510332.
- Jochinke, D. C., Noonon, B. J., Wachsmann, N. G., & Norton, R. M. (2007). The adoption of precision agriculture in an Australian broadacre cropping system—Challenges and opportunities. *Field Crops Research*, 104(1), 68-76, doi:10.1016/j.fcr.2007.05.016.
- Kim, H. J., Sudduth, K., A., Hummel, J., W., & Drummond, S., T. (2007). *Application of Ion-Selective Electrodes for Simultaneous Analysis of Soil Macronutrients*. Paper presented at the 2007 ASAE Annual Meeting, St. Joseph, Mich.,
- Kodaira, M., & Shibusawa, S. (2013). Using a mobile real-time soil visible-near infrared sensor for high resolution soil property mapping. *Geoderma*, 199, 64-79, doi:10.1016/j.geoderma.2012.09.007.
- Korsaeth, A. (2008). Dependence of Soil Apparent Electrical Conductivity (ECa) upon Soil Texture and Ignition Loss at Various Depths in Two Morainic Loam Soils in Southeast Norway. In B. J. Allred, J. J. Daniels, & M. R. Ehsani (Eds.), *Handbook of Agricultural Geophysics* (pp. 217-223, Books in Soils, Plants, and the Environment). Boca Raton, FL: CRC Press.
- Kuang, B., Mahmood, H. S., Quraishi, M. Z., Hoogmoed, W. B., Mouazen, A. M., & van Henten, E. J. (2012). Sensing Soil Properties in the Laboratory, In Situ, and On-Line: A Review. In D. L. Sparks (Ed.), *Advances in Agronomy* (Vol. 114, pp. 155-223): Academic Press.
- Kubitza, C., Krishna, V. V., Urban, K., Alamsyah, Z., & Qaim, M. (2018). Land Property Rights, Agricultural Intensification, and Deforestation in Indonesia. *Ecological Economics*, 147, 312-321, doi:10.1016/j.ecolecon.2018.01.021.

- La, W.-J., Sudduth, K. A., Kim, H.-J., & Chung, S.-O. (2016). Fusion of Spectral and Electrochemical Sensor Data for Estimating Soil Macronutrients. *59*(4), doi:10.13031/trans.59.11562.
- Lakshmi, V., Jackson, T. J., & Zehrhuhs, D. (2003). Soil moisture–temperature relationships: results from two field experiments. *Hydrological Processes*, *17*(15), 3041-3057, doi:10.1002/hyp.1275.
- Lee, K. S., Lee, D. H., Sudduth, K. A., Chung, S. O., Kitchen, N. R., & Drummond, S. T. (2009). Wavelength Identification and Diffuse Reflectance Estimation for Surface and Profile Soil Properties. *Transactions of the ASABE*, *52*(3), 683-695, doi:10.13031/2013.27385.
- Li, Y., Yang, Q., Chen, M., Wang, M., & Zhang, M. (2019). An ISE-based On-Site Soil Nitrate Nitrogen Detection System. *Sensors*, *19*(21), 4669.
- Liu, Y., Pan, X., & Li, J. (2015). Current Agricultural Practices Threaten Future Global Food Production. *Journal of Agricultural and Environmental Ethics*, *28*(2), 203-216, doi:10.1007/s10806-014-9527-6.
- Lobsey, C. R., Viscarra Rossel, R. A., & McBratney, A. B. (2010). Proximal Soil Nutrient Sensing Using Electrochemical Sensors. In R. A. Viscarra Rossel, A. B. McBratney, & B. Minasny (Eds.), *Proximal Soil Sensing* (pp. 77-88). Dordrecht: Springer Netherlands.
- Lowery, B., & Morrison, J. E. (2002). Soil Penetrometers and Penetrability. In J. H. Dane, & C. G. Topp (Eds.), *Methods of Soil Analysis: Part 4 Physical Methods* (pp. 363-388, SSSA Book Series, Vol. 5.4). Madison, WI: Soil Science Society of America.
- Lund, E., Adamchuk, V., Collings, K., Drummond, P., & Christy, C. (2005). Development of soil pH and lime requirement maps using on-the-go soil sensors. *Precision Agriculture 2005, ECPA 2005*.
- Mahmood, H. S., Hoogmoed, W. B., & van Henten, E. J. (2012). Sensor data fusion to predict multiple soil properties. [journal article]. *Precision Agriculture*, *13*(6), 628-645, doi:10.1007/s11119-012-9280-7.
- Mapare, S., Yu, P.-L., Sarkar, A., & Mukhopadhyay, S. C. (2013). *A Review of sensor technology for in-field phosphate monitoring*.
- Martinelli, L. A., & Filoso, S. (2008). Expansion of Sugarcane Ethanol Production in Brazil: Environmental and Social Challenges. *Ecological Applications*, *18*(4), 885-898, doi:10.1890/07-1813.1.
- Matson, P. A., Parton, W. J., Power, A. G., & Swift, M. J. (1997). Agricultural Intensification and Ecosystem Properties. *Science*, *277*(5325), 504-509, doi:10.1126/science.277.5325.504.
- McKyes, E. (1985). *Soil cutting and tillage* (Vol. 7): Elsevier.
- McNeill, J. D. (1980a). Electrical Conductivity of Soils and Rocks. *Technical Note 5*. Ontario, Canada: Geonics Limited.
- McNeill, J. D. (1980b). Electromagnetic Terrain Conductivity measurement at Low Induction Numbers. *Technical Note 6*. Ontario, Canada: Geonics Limited.
- Menció, A., Mas-Pla, J., Otero, N., Regàs, O., Boy-Roura, M., Puig, R., et al. (2016). Nitrate pollution of groundwater; all right..., but nothing else? *Science of The Total Environment*, *539*, 241-251, doi:10.1016/j.scitotenv.2015.08.151.
- Mester, A., van der Kruk, J., Zimmermann, E., & Vereecken, H. (2011). Quantitative Two-Layer Conductivity Inversion of Multi-Configuration Electromagnetic Induction Measurements. *Vadose Zone*, *10*(4), 1319-1330, doi:10.2136/vzj2011.0035.
- Miao, Z., Maohua, W., Li, C., Ang, S. S., Nguyen, C. V., & Jijun, Z. An automatic fluidic system for the rapid detection of soil nutrients. In *2008 IEEE International Conference on Automation and Logistics, 1-3 Sept. 2008 2008* (pp. 2742-2746). doi:10.1109/ICAL.2008.4636639.
- Michael, S., Gebbers, R., Eckart, K., & Jan, S. (2011). Soil pH Mapping with an On-The-Go Sensor. *Sensors*, *11*, 573-598, doi:10.3390/s110100573.
- Mulla, D. J. (2013). Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosystems Engineering*, *114*(4), 358-371, doi:10.1016/j.biosystemseng.2012.08.009.

- Mulvaney, R. L. (1996). Nitrogen—Inorganic Forms. In D. L. Sparks, A. L. Page, P. A. Helmke, & R. H. Loeppert (Eds.), *Methods of Soil Analysis Part 3—Chemical Methods* (pp. 1123-1184, SSSA Book Series, Vol. 5.3). Madison, WI: Soil Science Society of America, American Society of Agronomy.
- Nagendran, R. (2011). Chapter 24 - Agricultural Waste and Pollution. In T. M. Letcher, & D. A. Vallero (Eds.), *Waste* (pp. 341-355). Boston: Academic Press.
- Nise, N. S. (2015). *Control systems engineering* (7th ed.). Hoboken, NJ: Wiley.
- O'Connor, M. (2012). Precision Agriculture: Opportunities and Challenges. In L. A. Davis, P. K. Enge, & G. X. Gao (Eds.), *Global Navigation Satellite Systems: Report of a Joint Workshop of the National Academy of Engineering and the Chinese Academy of Engineering* (pp. 199-206). Washington, DC: The National Academies Press.
- Ogata, K. (2010). *Modern control engineering* (5th ed., Instrumentation and control series). Boston: Prentice-Hall.
- Oliver, M. A. (2010). An Overview of Geostatistics and Precision Agriculture. In M. A. Oliver (Ed.), *Geostatistical Applications for Precision Agriculture* (pp. 1-34). Dordrecht: Springer Netherlands.
- Padhi, J., & Misra, R. K. (2011). Sensitivity of EM38 in determining soil water distribution in an irrigated wheat field. *Soil and Tillage Research*, 117, 93-102, doi:10.1016/j.still.2011.09.003.
- Patitz, W. E., Brock, B. C., & Powell, E. G. (1995). Measurement of Dielectric and Magnetic Properties of Soil. (pp. 26). Albuquerque, New Mexico and Livermore, California: Sandia National Laboratories.
- Pedraera-Parrilla, A., Van De Vijver, E., Van Meirvenne, M., Espejo-Pérez, A. J., Giráldez, J. V., & Vanderlinden, K. (2016). Apparent electrical conductivity measurements in an olive orchard under wet and dry soil conditions: significance for clay and soil water content mapping. *Precision Agriculture*, 17(5), 531-545, doi:10.1007/s11119-016-9435-z.
- Peterson, J. C. (1982). Effects of pH upon nutrient availability in a commercial soilless root medium utilized for floral crop production. *Ohio State University and Ohio Research and Development Center*, 268, 16-19.
- Price, R. R., Hummel, J. W., Birrell, S. J., & Ahmad, I. S. (2003). Rapid Nitrate Analysis of Soil Cores Using ISFETs. *Transactions of the ASAE*, 46(3), 601, doi:10.13031/2013.13588.
- Reece, A. R. (1964). The Fundamental Equation of Earth-Moving Mechanics. *Proceedings of the Institution of Mechanical Engineers, Conference Proceedings*, 179(6), 16-22, doi:10.1243/pime_conf_1964_179_134_02.
- Rogovska, N., Laird, D. A., Chiou, C.-P., & Bond, L. J. (2019). Development of field mobile soil nitrate sensor technology to facilitate precision fertilizer management. *Precision Agriculture*, 20(1), 40-55, doi:10.1007/s11119-018-9579-0.
- Saey, T., De Smedt, P., Islam, M. M., Meerschman, E., Van De Vijver, E., Lehouck, A., et al. (2012a). Depth slicing of multi-receiver EMI measurements to enhance the delineation of contrasting subsoil features. *Geoderma*, 189–190, 514-521, doi:10.1016/j.geoderma.2012.06.010.
- Saey, T., De Smedt, P., Meerschman, E., Islam, M. M., Meeuws, F., Van De Vijver, E., et al. (2012b). Electrical Conductivity Depth Modelling with a Multireceiver EMI Sensor for Prospecting Archaeological Features. *ARP Archaeological Prospection*, 19(1), 21-30.
- Saey, T., Simpson, D., Vermeersch, H., Cockx, L., & Van Meirvenne, M. (2009). Comparing the EM38DD and DUALEM-21S Sensors for Depth-to-Clay Mapping. *Soil Science Society of America*, 73(1), 7-12, doi:10.2136/sssaj2008.0079.
- Saey, T., Simpson, D., Vitharana, U. W. A., Vermeersch, H., Vermang, J., & Van Meirvenne, M. (2008). Reconstructing the paleotopography beneath the loess cover with the aid of an electromagnetic induction sensor. *CATENA*, 74(1), 58-64, doi:10.1016/j.catena.2008.03.007.
- Sah, R. N. (1994). Nitrate-nitrogen determination—a critical review. *Communications in Soil Science and Plant Analysis*, 25(17-18), 2841-2869, doi:10.1080/00103629409369230.

- Santos, F. A. M. (2004). 1-D laterally constrained inversion of EM34 profiling data. *Journal of Applied Geophysics*, 56(2), 123-134, doi:10.1016/j.jappgeo.2004.04.005.
- Santos, F. A. M., Triantafilis, J., Bruzgulis, K. E., & Roe, J. A. E. (2010a). Inversion of Multiconfiguration Electromagnetic (DUALEM-421) Profiling Data Using a One-Dimensional Laterally Constrained Algorithm. *Vadose Zone*, 9(1), 117-125, doi:10.2136/vzj2009.0088.
- Santos, F. A. M., Triantafilis, J., Taylor, R. S., Holladay, S., & Bruzgulis, K. E. (2010b). Inversion of Conductivity Profiles from EM Using Full Solution and a 1-D Laterally Constrained Algorithm. *Journal of Environmental & Engineering Geophysics*, 15(3), 163-174, doi:10.2113/jeeg15.3.163.
- Sariyildiz, E., Yu, H., & Ohnishi, K. (2015). A Practical Tuning Method for the Robust PID Controller with Velocity Feed-Back. *Machines*, 3(3), 208-222, doi:10.3390/machines3030208.
- Sasaki, Y. (1989). Two-dimensional joint inversion of magnetotelluric and dipole-dipole resistivity data. *Geophysics*, 54(2), 254-262, doi:10.1190/1.1442649.
- Savci, S. (2012). Investigation of Effect of Chemical Fertilizers on Environment. *APCBEE Procedia*, 1, 287-292, doi:10.1016/j.apcbee.2012.03.047.
- Schoning, M. J., Gluck, O., Thust, M., Ali, M., Pahlavanpour, B., Eklund, M., et al. (1999). Composition Measurement. In J. G. Webster (Ed.), *The Measurement, Instrumentation and Sensors Handbook on CD-ROM*. Boca Raton, FL: CRC Press.
- Sethuramasamyraja, B., Adamchuk, V. I., Dobermann, A., Marx, D. B., Jones, D. D., & Meyer, G. E. (2008). Agitated soil measurement method for integrated on-the-go mapping of soil pH, potassium and nitrate contents. *Computers and Electronics in Agriculture*, 60(2), 212-225, doi:10.1016/j.compag.2007.08.003.
- Sethuramasamyraja, B., Adamchuk, V. I., Marx, D. B., Dobermann, A., Meyer, G. E., & Jones, D. D. (2007). Analysis of an Ion-Selective Electrode Based Methodology for Integrated On-The-Go Mapping of Soil pH, Potassium, and Nitrate Contents. *Transactions of the ASABE*, 50(6), 1927-1935, doi:10.13031/2013.24088.
- Shibusawa, S. (2006). Soil Sensors for Precision Farming. In A. Srinivasan (Ed.), *Handbook of precision agriculture : principles and applications* (pp. 57-85). Boca Raton: CRC Press, Taylor & Francis Group.
- Sibley, K. J., Astatkie, T., Brewster, G., Struik, P. C., Adsett, J. F., & Pruski, K. (2008). Field-scale validation of an automated soil nitrate extraction and measurement system. *Precision Agriculture*, 10(2), 162-174, doi:10.1007/s11119-008-9081-1.
- Sinfield, J. V., Fagerman, D., & Colic, O. (2010). Evaluation of sensing technologies for on-the-go detection of macro-nutrients in cultivated soils. *Computers and Electronics in Agriculture*, 70(1), 1-18, doi:10.1016/j.compag.2009.09.017.
- Skoog, D. A., West, D. M., Holler, F. J., & Crouch, S. R. (2014). *Fundamentals of analytical chemistry*: Cengage Learning.
- Smolka, M., Puchberger-Enengl, D., Bipoun, M., Klasa, A., Kiczakajlo, M., Śmiechowski, W., et al. (2017). A mobile lab-on-a-chip device for on-site soil nutrient analysis. *Precision Agriculture*, 18(2), 152-168, doi:10.1007/s11119-016-9452-y.
- Sparks, D. L., Soil Science Society of, A., & American Society of, A. (1996). *Methods of soil analysis. Part 3*. Madison, Wis.: Soil Science Society of America : American Society of Agronomy.
- Srinivasan, A. (2006). Precision Agriculture: An Overview. In A. Srinivasan (Ed.), *Handbook of precision agriculture : principles and applications* (pp. 3-15). Boca Raton: CRC Press, Taylor & Francis Group.
- Staff, S. S. (2014). Soil Survey Investigations Report No. 51 Version 2. In R. Burt (Ed.), *Soil Survey Field and Laboratory Methods Manual* (pp. 265). Nebraska: Department of Agriculture, Natural Resources Conservation Service.

- Sudarsan, B., Ji, W., Adamchuk, V., & Biswas, A. (2018). Characterizing soil particle sizes using wavelet analysis of microscope images. *Computers and Electronics in Agriculture*, *148*, 217-225, doi:10.1016/j.compag.2018.03.019.
- Sudduth, K. A., Myers, D. B., Kitchen, N. R., & Drummond, S. T. (2013). Modeling soil electrical conductivity–depth relationships with data from proximal and penetrating ECa sensors. *Geoderma*, *199*, 12-21, doi:10.1016/j.geoderma.2012.10.006.
- Sun, Y., Druucker, H., Hartung, E., Hueging, H., Cheng, Q., Zeng, Q., et al. (2011). Map-based investigation of soil physical conditions and crop yield using diverse sensor techniques. *Soil and Tillage Research*, *112*(2), 149-158, doi:10.1016/j.still.2010.12.002.
- Thomas, G. W. (1996). Soil pH and Soil Acidity. In D. L. Sparks, A. L. Page, P. A. Helmke, & R. H. Loeppert (Eds.), *Methods of Soil Analysis Part 3—Chemical Methods* (pp. 475-490, SSSA Book Series, Vol. 5.3). Madison, WI: Soil Science Society of America, American Society of Agronomy.
- Thottan, J., Adsett, J. F., Sibley, K. J., & MacLeod, C. M. (1994). Laboratory evaluation of the ion selective electrode for use in an automated soil nitrate monitoring system. *Communications in Soil Science and Plant Analysis*, *25*(17-18), 3025-3034, doi:10.1080/00103629409369243.
- Tirado, R. (2008). *Dead Zones : How Agricultural Fertilizers are Killing our Rivers, Lakes and Oceans*. Exeter, GB: Greenpeace Research Laboratories.
- Triantafilis, J., & Monteiro Santos, F. A. (2013). Electromagnetic conductivity imaging (EMCI) of soil using a DUALEM-421 and inversion modelling software (EM4Soil). *Geoderma*, *211–212*, 28-38, doi:10.1016/j.geoderma.2013.06.001.
- Triantafilis, J., Page, D., Wege, R., Wong, V., & Monteiro Santos, F. A. (2012). Modeling the electrical conductivity of hydrogeological strata using joint-inversion of loop-loop electromagnetic data. *Geophysics*, *77*(4).
- Triantafilis, J., Roe, J. A. E., & Monteiro Santos, F. A. (2011). Detecting a leachate plume in an aeolian sand landscape using a DUALEM-421 induction probe to measure electrical conductivity followed by inversion modelling. *Soil Use and Management*, *27*(3), 357-366, doi:10.1111/j.1475-2743.2011.00352.x.
- Triantafilis, J., & Santos, F. A. M. (2009). 2-Dimensional soil and vadose-zone representation using an EM38 and EM34 and a laterally constrained inversion model. *Australian Journal of Soil Research*, *47*.
- Triantafilis, J., & Santos, F. A. M. (2010). Resolving the spatial distribution of the true electrical conductivity with depth using EM38 and EM31 signal data and a laterally constrained inversion model. *Australian Journal of Soil Research*, *48*.
- Triantafilis, J., Terhune Iv, C. H., & Monteiro Santos, F. A. (2013). An inversion approach to generate electromagnetic conductivity images from signal data. *Environmental Modelling & Software*, *43*, 88-95, doi:10.1016/j.envsoft.2013.01.012.
- Truog, E. (1947). Soil reaction influence on availability of plant nutrients. *Soil Science Society of America*, *11*(C), 305-308.
- USDA (2012). Cone Penetrometer. In *Part 631 National Engineering Handbook* (pp. 13). United States: USDA NRCS.
- Van Alphen, B. J., & Stoorvogel, J. J. (2000). A Methodology for Precision Nitrogen Fertilization in High-Input Farming Systems. *Precision Agriculture*, *2*(4), 319-332, doi:10.1023/a:1012338414284.
- Van Noordwijk, M., Khasanah, N. m., & Dewi, S. (2017). Can intensification reduce emission intensity of biofuel through optimized fertilizer use? Theory and the case of oil palm in Indonesia. *GCB Bioenergy*, *9*(5), 940-952, doi:10.1111/gcbb.12398.
- Vasilescu, G. (2006). *Electronic noise and interfering signals: principles and applications*: Springer Science & Business Media.

- Viscarra Rossel, R. A., Adamchuk, V. I., Sudduth, K. A., McKenzie, N. J., & Lobsey, C. (2011). Proximal Soil Sensing: An Effective Approach for Soil Measurements in Space and Time. In D. L. Sparks (Ed.), *Advances in Agronomy* (Vol. 113, pp. 243-291): Academic Press.
- Viscarra Rossel, R. A., Gilbertson, M., Thylen, L., Hansen, O., McVey, S., & McBratney, A. B. Field measurements of soil pH and lime requirement using an on-the-go soil pH and lime requirement measurement system. In *Proceedings of the 5th European Conference on Precision Agriculture, Wageningen Academic Publishers, the Netherlands, 2005* (pp. 511-519)
- Viscarra Rossel, R. A., & Walter, C. (2004). Rapid, quantitative and spatial field measurements of soil pH using an Ion Sensitive Field Effect Transistor. *Geoderma*, *119*(1), 9-20, doi:10.1016/S0016-7061(03)00219-2.
- Vosti, S. A., Witcover, J., Carpentier, C. L., Oliveira, S. J. M. D., & Santos, J. C. D. (2001). Intensifying Small-scale Agriculture in the Western Brazilian Amazon: Issues, Implications and Implementation. In D. R. Lee, & C. B. Barrett (Eds.), *Tradeoffs or Synergies? : Agricultural Intensification, Economic Development, and the Environment* (pp. 245-266). Wallingford, Oxon, UK: CABI Publishing.
- Walter, J., Lück, E., Bauriegel, A., Richter, C., & Zeitz, J. (2015). Multi-scale analysis of electrical conductivity of peatlands for the assessment of peat properties. *European Journal of Soil Science*, *66*(4), 639-650, doi:10.1111/ejss.12251.
- Walter, J., Lück, E., Heller, C., Bauriegel, A., & Zeitz, J. (2019). Relationship between electrical conductivity and water content of peat and gyttja: implications for electrical surveys of drained peatlands. *Near Surface Geophysics*, *17*(2), 169-179, doi:10.1002/nsg.12030.
- Wang, X., Jiang, J., & Yang, H. (2011). An Interface Circuit for Ion-selective Electrode Sensors. *Procedia Engineering*, *15*, 2609-2613, doi:10.1016/j.proeng.2011.08.490.
- Watson, K. (2007). *Soils Illustrated - Field Descriptions First Edition*. Kamloops, B.C.: International Remote Sensing Surveys.
- Wheeler, A. J., & Ganji, A. R. (2004). General Characteristics of Measurement Systems. In *Introduction to Engineering Experimentation* (Vol. 3, pp. 14 - 15). Upper Saddle River, New Jersey: Pearson Higher Education.
- Wheeler, A. J., & Ganji, A. R. (2010). *Introduction to engineering experimentation* (Third ed.). Upper Saddle River, N.J.: Pearson Higher Education.
- Wienhold, B. J., & Doran, J. W. (2008). Apparent Electrical Conductivity for Delineating Spatial Variability in Soil Properties. In B. J. Allred, J. J. Daniels, & M. R. Ehsani (Eds.), *Handbook of Agricultural Geophysics* (pp. 211-215, Books in Soils, Plants, and the Environment): CRC Press.
- Yan, X. T., Bianco, A., Niu, C., Palazzetti, R., Henry, G., Li, Y., et al. (2020). The AgriRover: A Reinvented Mechatronic Platform from Space Robotics for Precision Farming. In X.-T. Yan, D. Bradley, D. Russell, & P. Moore (Eds.), *Reinventing Mechatronics: Developing Future Directions for Mechatronics* (pp. 55-73). Chambridge: Springer International Publishing.
- Yan, X. T., Donaldson, Karen M., Davidson, C. M., Gao, Y., Wu, H., Houston, A. M., et al. (2018). Effects of sample pretreatment and particle size on the determination of nitrogen in soil by portable LIBS and potential use on robotic-borne remote Martian and agricultural soil analysis systems. [10.1039/C8RA07065B]. *RSC Advances*, *8*(64), 36886-36894, doi:10.1039/C8RA07065B.
- Zare, E., Huang, J., Santos, F. A. M., & Triantafilis, J. (2015). Mapping Salinity in Three Dimensions using a DUALEM-421 and Electromagnetic Inversion Software. *Soil Science Society of America*, *79*(6), 1729-1740, doi:10.2136/sssaj2015.06.0238.
- Zhang, R., & Wienhold, B. J. (2002). The effect of soil moisture on mineral nitrogen, soil electrical conductivity, and pH. *Nutrient Cycling in Agroecosystems*, *63*(2), 251-254, doi:10.1023/A:1021115227884.

- Zhdanov, M. S. (2015). Ill-Posed Problems and the Methods of Their Solution. In M. S. Zhdanov (Ed.), *Inverse Theory and Applications in Geophysics (Second Edition)* (pp. 33-61). Oxford: Elsevier.
- Zhu, K. G., Ma, M. Y., Che, H. W., Yang, E. W., Ji, Y. J., Yu, S. B., et al. (2012). PC-based artificial neural network inversion for airborne time-domain electromagnetic data. *Applied Geophysics*, 9(1), 1-8, doi:10.1007/s11770-012-0307-7.

9. Appendices

9.1. Automatic Soil Analyzer Electronic Diagrams

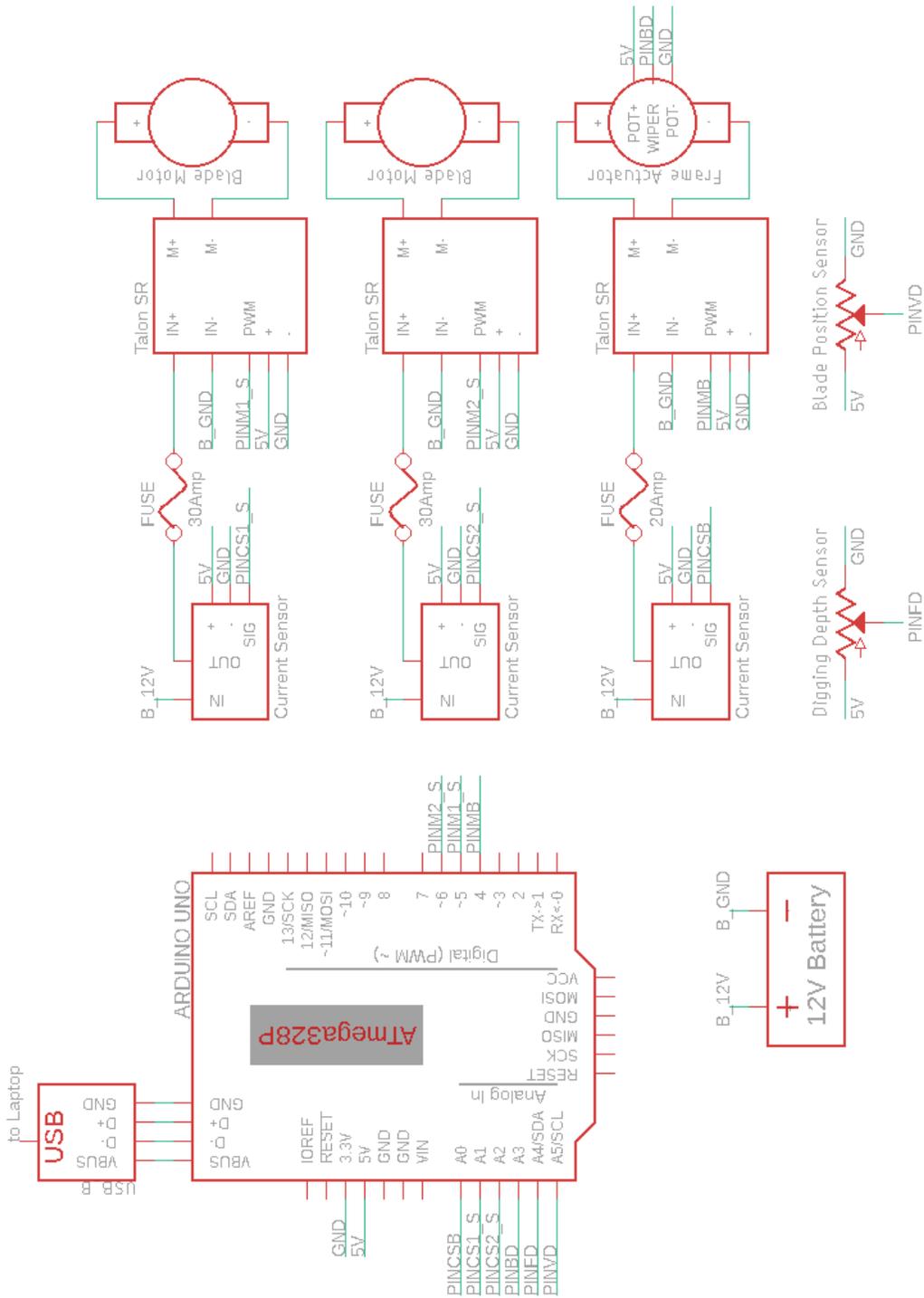


Figure 9-1. The OSA motors and actuators schematic wiring diagram

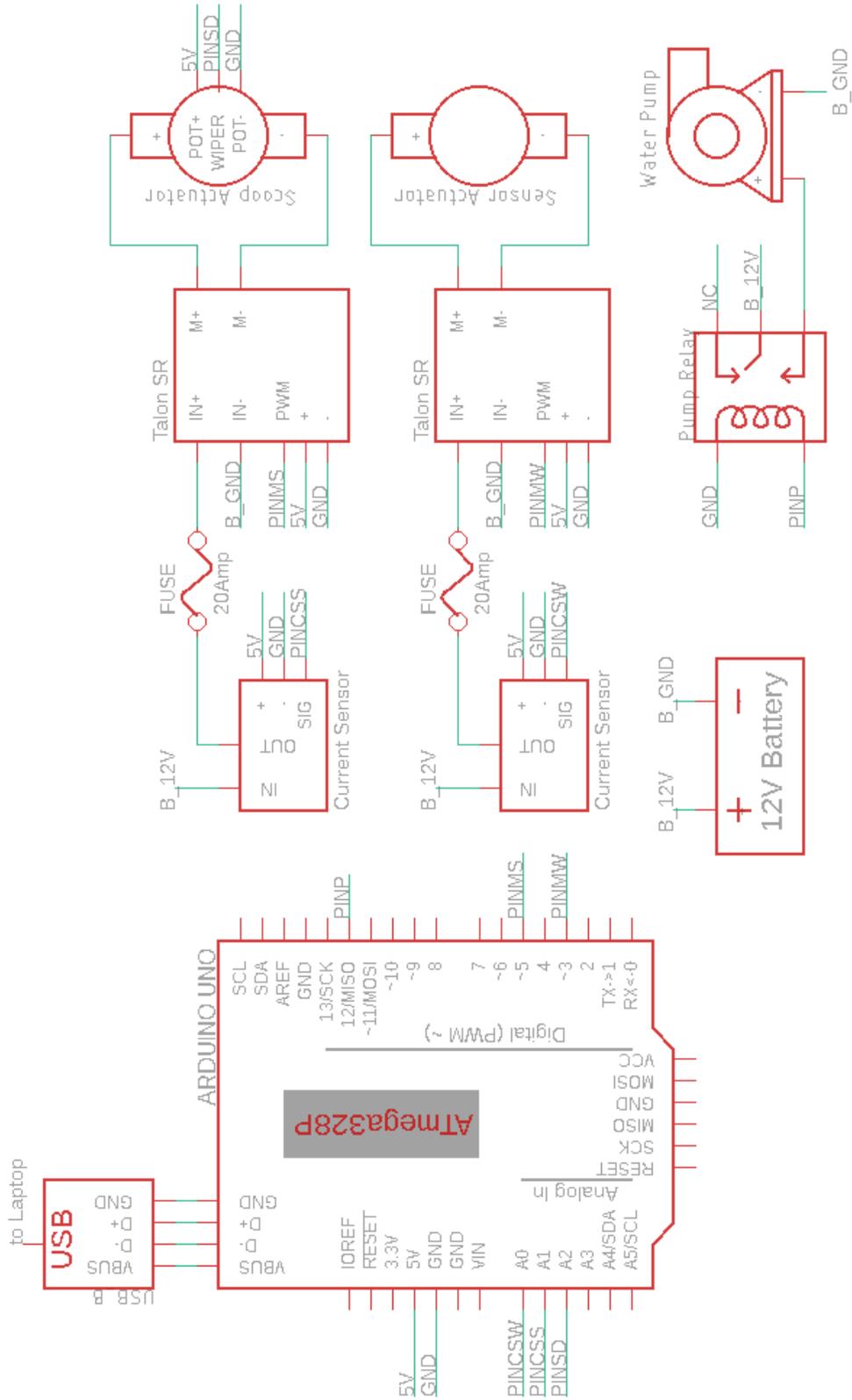


Figure 9-2. The OSA sensors schematic wiring diagram

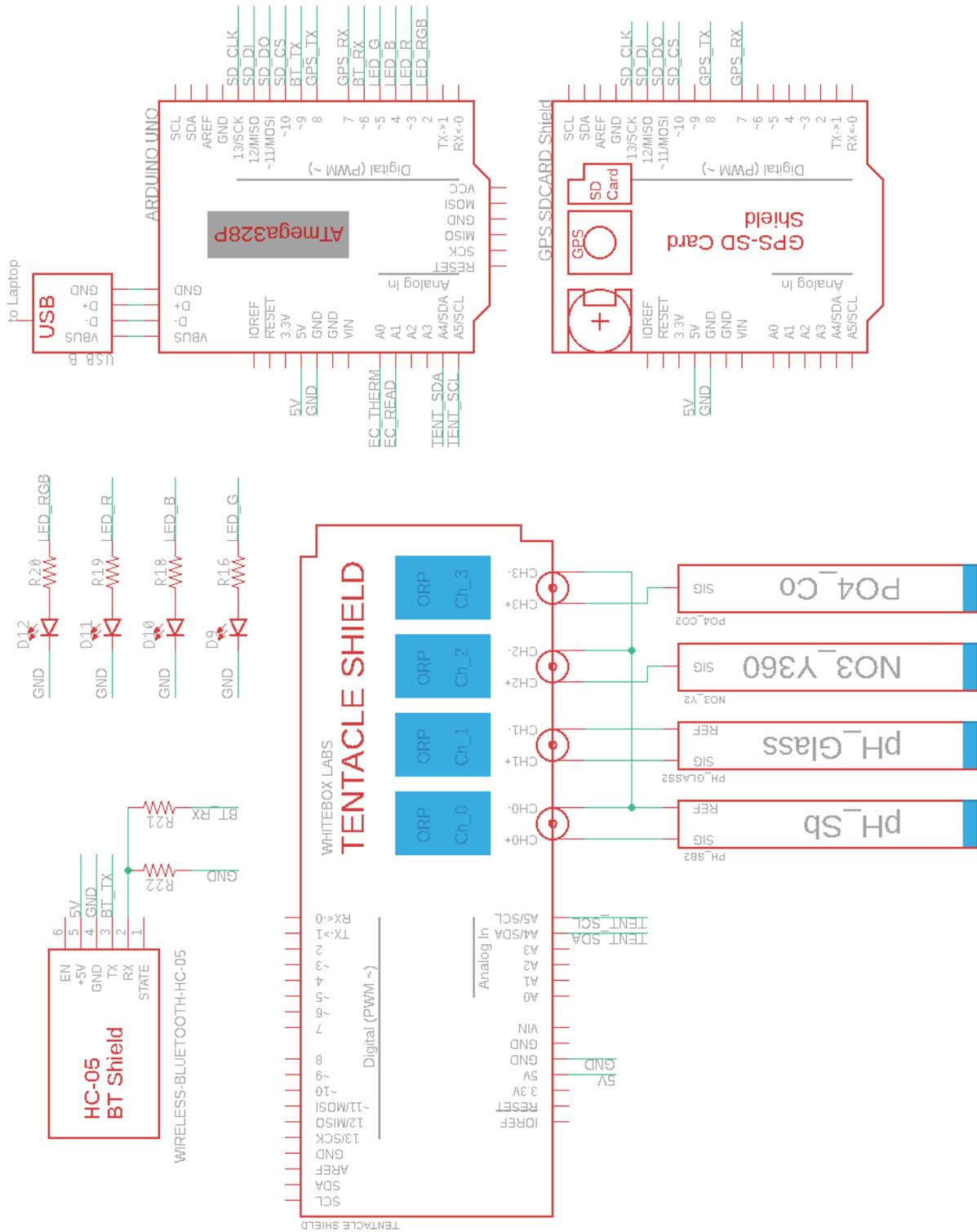


Figure 9-3. The OSA DAQ schematic wiring diagram

9.2. Automatic Soil Analyzer Control and DAQ Code

9.2.1. Digging Controller Code

```
//Ziegler Nichols Blade_Black_PID_LV_08-10-2018
#include <Servo.h>
#include "RunningMedian.h"
#include "RunningAverage.h"

//PWM pins
const int pinM1_S = 5;           // signal pin blademotor1
const int pinM2_S = 6;           // signal pin blademotor2
const int pinMB = 4;             // black motor pin
const int pinMS = 9;            // scoop motor pin
//analog pins
const int pinCSB = A0;           //pin check current black motor
const int pinCS1_S = A1;        //PIN current blade motor 1
const int pinCS2_S = A2;        //PIN current blade motor 2
const int pinBD = A3;           //pin check position black black
const int pinFD = A4;           //pin check position frame height
const int pinVD = A5;           //pin check blade motor position (max = 1023)

//constant
const int FLOAT_BUFFER_SIZE = 8; // float buffer for any calculation
const int CURRENT_BUFFER_SIZE = 32; // current buffer size
const int READ_BUFFER_SIZE = 128;
const int maxCurr = 30;          // maximum current
const int noSpeedBlade = 93;     // blade stop
const int slowSpeedBlade = 100; // blade slow speed for flat blade position
const int topSpeedBlade = 180;   // blade top speed
const int noSpeedBlack = 95;     // black motor stop
const int DATA_LENGTH = 128;

//VARIABLES :
//Input command
unsigned long current_time;
String message;                 // whole input message from user
char COM;                       // command "case" extract from input message
String VAL_s;                   // string value received from input message
int VAL_n;                      // convert Val_s to int
float current1ar;               //current blade motor 1,2, black motor
float current2ar;
int currentBar;
unsigned long startTime;        // count time White motor start to go down
unsigned long finishTime;       // count time White motor at the bottom
int sum;
float I;                        // current Blade motor in Amphere for output print https://goo.gl/8LT7Rk
float I_raw;                   // current Blade motor raw

//Blade
int cSpeedBlade;                // BLADE motor speed
int pos_raw_Blade;              // analog read position sensor (max 1023)
int deg_Blade;                  // convert pos_raw_Blade to degree (flat = 0, 180)
char Amp_1[FLOAT_BUFFER_SIZE]; // double to string float current1 (current blade motor1)
char Amp_2[FLOAT_BUFFER_SIZE]; // double to string float current2 (current blade motor2)
char current_buffer[CURRENT_BUFFER_SIZE]; // current blade motor buffer

//Black motor
char Amp_B[FLOAT_BUFFER_SIZE]; // double to string float current3 (current black motor)
char h_B[FLOAT_BUFFER_SIZE];   // sprintf for height black motor
char h_F[FLOAT_BUFFER_SIZE];   // sprintf for height frame
int cSpeedBlack;               // BLACK motor speed

int pos_raw_Black;             // analog read position black motor
int height_Black;              // height conversion from analog read position black sensor
int last_pos_Black;            // height of last black motor position
```

```

int pos_raw_Frame;           // analog read position frame
int depth_Frame;            // height conversion from analog read position frame
int last_pos_Frame;         // height of last frame position (digging depth)

bool flag;                  // PID on/off
//for current conversion we need float
float l_raw1;
float l_raw2;
float l1;
float l2;
int amp_raw;
int amp;
int amp_a;
int amp_m;
int rate;
int rate_a;    //average rate
int rate_m;    //median rate
int er;
int er_a;
int er_m;
int offset = 95;           // offset 95 = no speed black
const int setpoint = 5;    // blade current amperage PID setpoint
float k_p = 9;             //optimal PID : 9 0 2
float k_i = 0;
float k_d = 2;
const int N_SAMPLES = 5;  //5samples of Blade current measurement ; aslinya 5
RunningAverage error(N_SAMPLES);
RunningAverage error_a(N_SAMPLES);
RunningAverage error_m(N_SAMPLES);
RunningAverage amp_average(N_SAMPLES);
RunningMedian amp_median(N_SAMPLES);
RunningMedian dig(N_SAMPLES);
char rate_s[FLOAT_BUFFER_SIZE];
char rate_avg[FLOAT_BUFFER_SIZE];
char rate_med[FLOAT_BUFFER_SIZE];
char amp_b[FLOAT_BUFFER_SIZE];
char amp_avg[FLOAT_BUFFER_SIZE];
char amp_med[FLOAT_BUFFER_SIZE];
char kp_s[FLOAT_BUFFER_SIZE];
char ki_s[FLOAT_BUFFER_SIZE];
char kd_s[FLOAT_BUFFER_SIZE];
char offset_s[FLOAT_BUFFER_SIZE];
char setpoint_s[FLOAT_BUFFER_SIZE];
char error_s[FLOAT_BUFFER_SIZE];
char error_avg[FLOAT_BUFFER_SIZE];
char error_med[FLOAT_BUFFER_SIZE];
char depth[FLOAT_BUFFER_SIZE];
char read_buffer[READ_BUFFER_SIZE];
int d_frame;               //desired depth of digging from user input
const int travel_h = 44;   //black motor height start when M pressed
int dig_h;                 //reading median dig

Servo m1;                 // Blade motor1
Servo m2;                 // Blade motor2
Servo mB;                 // Black motor

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  m1.attach(pinM1_S);      // set pin Blade motor1
  m2.attach(pinM2_S);      // set pin Blade motor2
  mB.attach(pinMB);        // set pin Black motor
  m1.write(noSpeedBlade);
  m2.write(noSpeedBlade);
  mB.write(noSpeedBlack);
}

```

```

Serial.println(F("Time ; rate ; rate_avg ; rate_med ; amp_raw ; amp_avg ; amp_med ; offset ; set point ; kp ; ki ; kd ; error ; er_a ; er_m ; depth ;
Checksum"));
}

```

```

void loop() {
if (Serial.available() > 0) {
message = " ";
message = Serial.readString();
COM = message.charAt(0);
VAL_s = message.substring(1, 5); // 90-180 knob for cutter speed, 0-90 knob for blade position
VAL_n = VAL_s.toInt();
switch (COM) {
case 'A': //Move up and down black motor https://goo.gl/DEQ9eY ; >95 up ; <95 down
cSpeedBlack = VAL_n;
mB.write(cSpeedBlack);
if (cSpeedBlack > 95) {
Serial.println("Frame goes up");
Serial.println("Height ; Current");
}
else {
Serial.println("Frame goes down");
Serial.println("Height ; Current");
}
break;
case 'E': //Stop black motor
mB.write(noSpeedBlack);
Serial.print("Height Black Motor : ");
Serial.println(checkHeight());
break;
case 'P': //Check height black motor and digging (frame) depth
Serial.print("Height Black Motor : ");
Serial.println(checkHeight());
Serial.print("Digging Depth : ");
Serial.println(checkDig());
break;
case 'O': //Check blade position
Serial.print("Blade Position : ");
Serial.println(checkPosition());
break;
case 'I': //Check current blade and black motor
current1ar = checkCurrent(1);
current2ar = checkCurrent(2);
currentBar = checkCurrent(3);
dtostrf(current1ar, 3, 0, Amp_1); // dtostrf (variable dibaca, width of string = 4 , precision = 1, saved to) XX.X
dtostrf(current2ar, 3, 0, Amp_2);
dtostrf(currentBar, 3, 0, Amp_B);
sprintf(current_buffer, "%s;%s;%s", Amp_1, Amp_2, Amp_B); // sprintf(buffer tempat ngesave, format, buffer yang dibaca)
Serial.print("Current: ");
Serial.println(current_buffer);
break;
case 'Q': // rotate blade manually 90-180 knob for cutter speed
cSpeedBlade = VAL_n;
m1.write(cSpeedBlade);
m2.write(cSpeedBlade);
break;
case 'R': // slowly start Blade motor
cSpeedBlade = noSpeedBlade;
while (cSpeedBlade < topSpeedBlade) {
cSpeedBlade++;
m1.write(cSpeedBlade);
m2.write(cSpeedBlade);
delay(100); //delay supaya pelan2, kalo tidak ada delay = HIGH CURRENT
}
break;
case 'U': //Hard Stop blade motor
m1.write(noSpeedBlade);
}
}
}

```

```

m2.write(noSpeedBlade);
Serial.println(checkPosition());
break;
case 'V': //Soft stop blade motor
while (cSpeedBlade > noSpeedBlade) {
  cSpeedBlade--;
  m1.write(cSpeedBlade);
  m2.write(cSpeedBlade);
  Serial.println(checkPosition());
  delay(20);
}
break;
case 'S': //flatten blade
deg_Blade = checkPosition();
while (deg_Blade > 330 || deg_Blade < 315) { //Move blade to flat position ; add slack due to chain-sprocket backlash
  m1.write(slowSpeedBlade);
  m2.write(slowSpeedBlade);
  deg_Blade = checkPosition();
  delay(10); // make more delay so position sensor has time, result wont -456 something
}
m1.write(noSpeedBlade);
m2.write(noSpeedBlade);
Serial.println(checkPosition());
break;
case 'M': //initialize condition : stop blade, move black motor up until reaches top
mB.write(140);
delay(1500);
mB.write(115);
delay(3000); //tadinya 5000
last_pos_Black = checkHeight();
while (last_pos_Black < travel_h) {
  mB.write(140);
  last_pos_Black = checkHeight();
  delay(10);
}
mB.write(noSpeedBlack);
Serial.println("Frame at top and Blade stopped, ready to go!");
Serial.flush();
break;
case 'C': //PID mode
cSpeedBlade = noSpeedBlade; // soft start Blade motor
while (cSpeedBlade < topSpeedBlade) {
  cSpeedBlade++;
  m1.write(cSpeedBlade);
  m2.write(cSpeedBlade);
  delay(50);
  mB.write(87);
}
flag = true; // Turn PID ON
d_frame = VAL_n; // d_frame = 2 (mm) cukup
while (flag == true) {
  amp = amp_blade(); //call for addvalue
  amp_a = amp_average.getAverage();
  amp_m = amp_median.getMedian();
  if (Serial.available() > 0) { //interrupt
    message = " ";
    message = Serial.readString();
    COM = message.charAt(0);
    switch (COM) {
      case 'E': //Stop black motor + blade
        mB.write(noSpeedBlack);
        Serial.print("Height Black Motor : ");
        Serial.println(checkHeight());
        flag = false;
        m1.write(noSpeedBlade);
        m2.write(noSpeedBlade);

```

```

        Serial.println("Interrupted");
        break;
    }
    break;
}
rate = offset + k_p * (amp - setpoint) + k_i * error.getAverage() + k_d * ((amp - setpoint) - error.getAverage());
rate_a = offset + k_p * (amp_a - setpoint) + k_i * error_a.getAverage() + k_d * ((amp_a - setpoint) - error_a.getAverage());
rate_m = offset + k_p * (amp_m - setpoint) + k_i * error_m.getAverage() + k_d * ((amp_m - setpoint) - error_m.getAverage());
error.addValue(amp - setpoint);
error_a.addValue(amp_a - setpoint);
error_m.addValue(amp_m - setpoint);
er = error.getAverage();
er_a = error_a.getAverage();
er_m = error_m.getAverage();
mB.write(rate_m);
last_pos_Frame = checkDig();
dig_h = dig.getMedian();
dtostrf(rate, 3, 0, rate_s);
dtostrf(rate_a, 3, 0, rate_avg);
dtostrf(rate_m, 3, 0, rate_med);
dtostrf(amp, 3, 0, amp_b);
dtostrf(amp_a, 3, 0, amp_avg);
dtostrf(amp_m, 3, 0, amp_med);
dtostrf(offset, 3, 0, offset_s);
dtostrf(setpoint, 3, 0, setpoint_s);
dtostrf(k_p, 3, 1, kp_s);
dtostrf(k_i, 3, 1, ki_s);
dtostrf(k_d, 3, 1, kd_s);
dtostrf(er, 3, 1, error_s);
dtostrf(er_a, 3, 1, error_avg);
dtostrf(er_m, 3, 1, error_med);
dtostrf(dig_h, 4, 1, depth);
current_time = millis();
sprintf(read_buffer, "%li;%s;%s;%s;%s;%s;%s;%s;%s;%s;%s;%s;%s", current_time, rate_s, rate_avg, rate_med, amp_b, amp_avg,
amp_med, offset_s, setpoint_s, kp_s, ki_s, kd_s, error_s, error_avg, error_med, depth);
int sum = checksum(read_buffer);
Serial.print(read_buffer); Serial.print(F("; chk: ")); Serial.println(sum);
if (dig_h > d_frame) { // interrupt + frame height, kalo ud sampe depth tertentu berhenti
    flag = false;
    mB.write(noSpeedBlack);
}
}
Serial.flush();
break;
case 'L': // setup after PID finished digging
    mB.write(140); // Frame up >95 up
    delay(1000);
    mB.write(noSpeedBlack);
    // soft stop Blade motor
    while (cSpeedBlade > noSpeedBlade) {
        cSpeedBlade--;
        m1.write(cSpeedBlade);
        m2.write(cSpeedBlade);
        Serial.println(checkPosition());
        delay(20);
    }
    delay(30);
    //Move blade to flat position ; add slack due to chain-sprocket backlash
    deg_Blade = checkPosition();
    while (deg_Blade > 330 || deg_Blade < 315) {
        m1.write(slowSpeedBlade);
        m2.write(slowSpeedBlade);
        deg_Blade = checkPosition();
        delay(20); // make more delay so position sensor has time, result wont -456 something
    }
    m1.write(noSpeedBlade);

```

```

m2.write(noSpeedBlade);
//move Black motor down until reach last position of digging
dig.clear();
last_pos_Frame = checkDig();
dig.add(last_pos_Frame);
dig_h = dig.getMedian();
d_frame = VAL_n;
while (dig_h < d_frame ) { // interrupt + frame height, kalo ud sampe depth tertentu berhenti
//   while (depth_Frame < d_frame) {
mB.write(75); // < 95 down
last_pos_Frame = checkDig();
dig.add(last_pos_Frame);
dig_h = dig.getMedian();
currentBar = checkCurrent(3);
Serial.println(dig_h);
delay(10); //need delay to stability, read sensor, calculate median. otherwise doesnot work
if (Serial.available() > 0 || currentBar > maxCurr) {
mB.write(noSpeedBlack);
Serial.println("interrupted or HIGH current");
break;
}
}
delay(300); // add delay to firmly contact soil
mB.write(noSpeedBlack);
Serial.println("Ready to deploy ISEs");
delay(100);
Serial.println("Now deploying ISEs");
Serial.flush();
break;
case 'X': //MCL
//M
m1.write(noSpeedBlade);
m2.write(noSpeedBlade);
last_pos_Black = checkHeight();
while (last_pos_Black < travel_h) {
mB.write(140);
last_pos_Black = checkHeight();
}
mB.write(noSpeedBlack);
//C
cSpeedBlade = noSpeedBlade; // soft start Blade motor
while (cSpeedBlade < topSpeedBlade) {
cSpeedBlade++;
m1.write(cSpeedBlade);
m2.write(cSpeedBlade);
delay(50);
mB.write(87);
}
flag = true; // Turn PID ON
d_frame = VAL_n; // d_frame = 2 mm cukup
while (flag == true) {
amp = amp_blade(); //call for addvalue
amp_a = amp_average.getAverage();
amp_m = amp_median.getMedian();
if (Serial.available() > 0) { //interrupt
message = " ";
message = Serial.readString();
COM = message.charAt(0);
switch (COM) {
case 'E': //Stop black motor + blade
mB.write(noSpeedBlack);
Serial.print("Height Black Motor : ");
Serial.println(checkHeight());
flag = false;
m1.write(noSpeedBlade);
m2.write(noSpeedBlade);

```

```

        Serial.println("Interrupted");
        break;
    }
    break;
}
rate = offset + k_p * (amp - setpoint) + k_i * error.getAverage() + k_d * ((amp - setpoint) - error.getAverage());
rate_a = offset + k_p * (amp_a - setpoint) + k_i * error_a.getAverage() + k_d * ((amp_a - setpoint) - error_a.getAverage());
rate_m = offset + k_p * (amp_m - setpoint) + k_i * error_m.getAverage() + k_d * ((amp_m - setpoint) - error_m.getAverage());
error.addValue(amp - setpoint);
error_a.addValue(amp_a - setpoint);
error_m.addValue(amp_m - setpoint);
er = error.getAverage();
er_a = error_a.getAverage();
er_m = error_m.getAverage();
mB.write(rate_m);
last_pos_Frame = checkDig();
dig_h = dig.getMedian();
dtostrf(rate, 3, 0, rate_s);
dtostrf(rate_a, 3, 0, rate_avg);
dtostrf(rate_m, 3, 0, rate_med);
dtostrf(amp, 3, 0, amp_b);
dtostrf(amp_a, 3, 0, amp_avg);
dtostrf(amp_m, 3, 0, amp_med);
dtostrf(offset, 3, 0, offset_s);
dtostrf(setpoint, 3, 0, setpoint_s);
dtostrf(k_p, 3, 1, kp_s);
dtostrf(k_i, 3, 1, ki_s);
dtostrf(k_d, 3, 1, kd_s);
dtostrf(er, 3, 1, error_s);
dtostrf(er_a, 3, 1, error_avg);
dtostrf(er_m, 3, 1, error_med);
dtostrf(dig_h, 4, 1, depth);
current_time = millis();
sprintf(read_buffer, "%li;%s;%s;%s;%s;%s;%s;%s;%s;%s;%s;%s;%s", current_time, rate_s, rate_avg, rate_med, amp_b, amp_avg,
amp_med, offset_s, setpoint_s, kp_s, ki_s, kd_s, error_s, error_avg, error_med, depth);
int sum = checksum(read_buffer);
Serial.print(read_buffer); Serial.print(F(" ; chk: ")); Serial.println(sum);
if (dig_h > d_frame ) { // frame height, kalo ud sampe depth tertentu berhenti
    flag = false;
    //L
    mB.write(140); // Frame up >95 up
    delay(1000);
    mB.write(noSpeedBlack);
    delay(10);
    // soft stop Blade motor
    while (cSpeedBlade > noSpeedBlade) {
        cSpeedBlade--;
        m1.write(cSpeedBlade);
        m2.write(cSpeedBlade);
        delay(20);
    }
    //Move blade to flat position ; add slack due to chain-sprocket backlash
    delay(30);
    deg_Blade = checkPosition();
    while (deg_Blade > 330 || deg_Blade < 315) {
        m1.write(slowSpeedBlade);
        m2.write(slowSpeedBlade);
        deg_Blade = checkPosition();
        delay(20); // make more delay so position sensor has time, result wont -456 something
    }
    m1.write(noSpeedBlade);
    m2.write(noSpeedBlade);
    //move Black motor down until reach last position of digging
    dig.clear();
    last_pos_Frame = checkDig();
    dig.add(last_pos_Frame);
}

```

```

dig_h = dig.getMedian();
while (dig_h < d_frame) { // interrupt + frame height, kalo ud sampe depth tertentu berhenti
// while (depth_Frame < d_frame) {
mB.write(75); // < 95 down
last_pos_Frame = checkDig();
dig.add(last_pos_Frame);
dig_h = dig.getMedian();
currentBar = checkCurrent(3);
delay(10); //need delay to stability, read sensor, calculate median. otherwise doesnot work
if (Serial.available() > 0 || currentBar > maxCurr) {
mB.write(noSpeedBlack);
Serial.println("interrupted or HIGH current");
break;
}
}
delay(200); // add delay to firmly contact soil
mB.write(noSpeedBlack);
Serial.println("Ready to deploy ISEs");
delay(2000); // add delay to observe blade is flat? cancel if not flat
Serial.println("Now deploying ISEs");
Serial.flush();
}
}
break;
}
}

//if manual blade selected
if (COM == 'Q' || COM == 'R') { // Q:rotate blade ; R:Slow start blade
current1ar = checkCurrent(1);
current2ar = checkCurrent(2);
dtostrf(current1ar, 3, 1, Amp_1); // dtostrf (variable dibaca, width of string = 4 , precision = 1, saved to) XX.X
dtostrf(current2ar, 3, 1, Amp_2);
sprintf(current_buffer, "%s;%s", Amp_1, Amp_2); // sprintf(buffer tempat ngesave, format, buffer yang dibaca)
Serial.print("Current: ");
Serial.println(current_buffer);
}
//safety current condition (> maxCurr --> stop blade and black motor)
if (checkCurrent(1) > maxCurr || checkCurrent(2) > maxCurr || checkCurrent(3) > maxCurr) {
Serial.println("High Current");
m1.write(noSpeedBlade);
m2.write(noSpeedBlade);
mB.write(noSpeedBlack);
}
//if manual black motor selected
if (COM == 'A') {
last_pos_Frame = checkDig();
dig.add(last_pos_Frame);
dig_h = dig.getMedian();
Serial.print(dig_h);
Serial.print(";");
Serial.print(checkHeight());
Serial.print(";");
Serial.println(checkCurrent(3));
}
Serial.flush();
}

int checkHeight() {
pos_raw_Black = analogRead(pinBD);
height_Black = (0.0194 * pos_raw_Black - 0.5237) * 2.54; // in inch, convert to cm
delay(10);
return height_Black;
}

int checkDig(void) { //in mm
pos_raw_Frame = analogRead(pinFD); // https://goo.gl/5gDVj9

```

```

depth_Frame = ((-0.0426 * pos_raw_Frame) + 32.608) * 10; // calibration D:\mcgill\OSA\Linear actuator - Pump\Calib_Frame_Height
delay(10);
dig.add(depth_Frame);
return depth_Frame;
}
int checkPosition() {
  pos_raw_Blade = analogRead(pinVD); // read positon sensor raw bit data. Arduino convert real voltage value to 0 - 1023
  deg_Blade = -0.3509 * pos_raw_Blade + 359; // convert to degree (linear calibration)
  delay(10); // kasi delay supaya stabil
  return deg_Blade;
}
float checkCurrent(int Wmotor) { //check current W(hich)motor
  switch (Wmotor) {
    case 1:
      I_raw = analogRead(pinCS1_S);
      break;
    case 2:
      I_raw = analogRead(pinCS2_S);
      break;
    case 3:
      I_raw = analogRead(pinCSB);
      break;
  }
  I = (I_raw - 510) * 5 / 1024 / 0.04 - 0.04; // Data processing:510-raw data from analogRead when the input is 0;
  // 5-5v; the first 0.04-0.04V/A(sensitivity); the second 0.04-offset val; current in A
  // https://goo.gl/8LT7Rk
  delay(30);
  return I;
}

// Checksum
int checksum(char* buf) {
  int sum = 0;
  for (int i = 0; i < DATA_LENGTH; i++) {
    sum = sum + buf[i];
  }
  return sum % 256;
}

int amp_blade(void) {
  I_raw1 = analogRead(pinCS1_S);
  I1 = (I_raw1 - 510) * 5 / 1024 / 0.04 - 0.04;
  I_raw2 = analogRead(pinCS2_S);
  I2 = (I_raw2 - 510) * 5 / 1024 / 0.04 - 0.04;
  amp_raw = (I1 + I2) / 2.0;
  amp_average.addValue(amp_raw);
  amp_median.add(amp_raw);
  return amp_raw;
}

```

9.2.2. ISEs Deployment Controller Code

```
//CLEAN for LV Scoop_White_ISE_LV_01-09-2018
#include <Servo.h> //Other than the Mega, use of the library disables analogWrite() (PWM) on pins 9 and 10
//whether or not there is a Servo on those pins. https://goo.gl/3s9aIX
//PWM pins : 3,5,6,9,10,11
//constant
const int pinMW = 3;           //white motor pin
const int pinMS = 5;           //scoop motor pin
const int pinCSW = A0;         //pin check current white motor
const int pinCSS = A1;         //pin check current black motor
const int pinSD = A2;         //pin check position scoop motor
const int pinP = 12;           // pin relay pump
const int noSpeedWhite = 95;   // servo stop
const int noSpeedScoop = 95;   // scoop motor stop
const int FLOAT_BUFFER_SIZE = 6; // float buffer for any calculation
const int CURRENT_BUFFER_SIZE = 20; // current buffer size
const int maxCurr = 10;        // maximum current

//variables
String message;
char COM;                       // command
String VAL_s;
int VAL_n;                       // value
int cSpeedWhite;                 // servo speed white motor
int cSpeedScoop;                 // servo speed scoop motor
int currentW;                     // current sensor white motor
int currentS;                     // current sensor scoop black motor
char Amp_W[FLOAT_BUFFER_SIZE];
char Amp_S[FLOAT_BUFFER_SIZE];
char current_buffer[CURRENT_BUFFER_SIZE]; // current buffer
float I;                          // current in Ampere https://goo.gl/8LT7Rk
float I_raw;                       // current raw
unsigned long startTime;           // count time White motor start to go down
unsigned long finishTime;         // count time White motor at the bottom
bool flag;
int pos_raw_Scoop;
int pos_Scoop;

Servo mW;
Servo mS;

void setup() {
  Serial.begin(9600);
  mW.attach(pinMW);
  mS.attach(pinMS);
  pinMode(pinP, OUTPUT);
  digitalWrite(pinP, LOW);
}

void loop() {
  if (Serial.available() > 0) {
    message = "";
    message = Serial.readString();
    COM = message.charAt(0);
    VAL_s = message.substring(1, 5); // 90-180 knob for cutter speed, 0-90 knob for blade position
    VAL_n = VAL_s.toInt();
    switch (COM) {
      case 'W': //Move up and down white motor manual https://goo.gl/DEQ9eY //>95 down //<95 up
        cSpeedWhite = VAL_n;
        mW.write(cSpeedWhite);
        break;
      case 'B': //Move up scoop motor >180 SOLID GREEN --> full forward PWM (>95) ; Position at 420 = top
        pos_Scoop = checkHeight();
        mS.write(180);
        while (pos_Scoop > 420) {
```

```

currentS = checkCurrent(2);
// Serial.print("Current Scoop motor : ");
// Serial.print(currentS);
pos_Scoop = checkHeight();
// Serial.print(" ; Position Scoop motor : ");
// Serial.println(pos_Scoop);
}
mS.write(noSpeedScoop);
break;
case 'D': //Move down scoop motor 6B SOLID RED --> full reverse PWM ; Position at 600 = bottom
pos_Scoop = checkHeight();
mS.write(6);
while (pos_Scoop < 600) {
currentS = checkCurrent(2);
// Serial.print("Current Scoop motor : ");
// Serial.print(currentS);
pos_Scoop = checkHeight();
// Serial.print(" ; Position Scoop motor : ");
// Serial.println(pos_Scoop);
}
mS.write(noSpeedScoop);
break;
case 'N': //Stop white +scoop motor manual
mW.write(noSpeedWhite);
mS.write(noSpeedScoop);
Serial.println("White & black motor stopped");
break;
case 'J': // motor white turun auto
startTime = millis();
finishTime = startTime + 3000;
mW.write(140);
while (millis() < finishTime) {
currentW = checkCurrent(1);
Serial.print("Current white motor : ");
Serial.println(currentW);
}
mW.write(noSpeedWhite);
break;
case 'F': // motor white naik auto
startTime = millis();
finishTime = startTime + 3000;
mW.write(40);
while (millis() < finishTime) {
currentW = checkCurrent(1);
Serial.print("Current white motor : ");
Serial.println(currentW);
}
mW.write(noSpeedWhite);
break;
case 'H': //pump on manual
digitalWrite(pinP, HIGH);
break;
case 'Y': //pump off manual
digitalWrite(pinP, LOW);
break;
case 'G': //pump auto on/off
startTime = millis();
finishTime = startTime + 4000;
while (millis() < finishTime) {
digitalWrite(pinP, HIGH);
}
digitalWrite(pinP, LOW);
break;
case 'K': //auto white down, hold, up, spray
//down
flag = true;

```

```

if (Serial.available() > 0) {
  message = " ";
  message = Serial.readString();
  COM = message.charAt(0);
  switch (COM) {
    case 'N':
      flag = false;
      mS.write(noSpeedScoop);
      mW.write(noSpeedWhite);
      Serial.println("Cancel Deploy ISEs");
      break;
    }
  }
  break;
}
if (flag == true) {
  startTime = millis();
  finishTime = startTime + 3000;
  mW.write(140);
  while (millis() < finishTime) {
    currentW = checkCurrent(1);           // call function to sense Amphere blade motor 2
    if (currentW > maxCurr) {
      flag = false;
      mW.write(noSpeedWhite);
      Serial.println("HIGH current");
      break;
    }
  }
  mW.write(noSpeedWhite);
  //hold ISE reading
  delay(30000);
  //up
  startTime = millis();
  finishTime = startTime + 3000;
  mW.write(40);
  while (millis() < finishTime) {
    currentW = checkCurrent(1);           // call function to sense Amphere blade motor 2
    if (currentW > maxCurr) {
      flag = false;
      mW.write(noSpeedWhite);
      Serial.println("HIGH current");
      break;
    }
  }
  mW.write(noSpeedWhite);
  Serial.println("ISEs measurement finished!");
  // delay(1);
  Serial.println("Cleaning up the ISEs");
  Serial.flush();
  //SCOOP down
  pos_Scoop = checkHeight();
  delay(1000); //delay for command M
  mS.write(6);
  while (pos_Scoop < 600) {
    currentS = checkCurrent(2);
    // Serial.print("Current Scoop motor : ");
    // Serial.print(currentS);
    pos_Scoop = checkHeight();
    // Serial.print(" ; Position Scoop motor : ");
    // Serial.println(pos_Scoop);
  }
  mS.write(noSpeedScoop);
  //pump on
  startTime = millis();
  finishTime = startTime + 3000;
  while (millis() < finishTime) {
    digitalWrite(pinP, HIGH);
  }
}

```

```

    }
    digitalWrite(pinP, LOW);
    //scoop up
    pos_Scoop = checkHeight();
    mS.write(180);
    while (pos_Scoop > 420) {
        currentS = checkCurrent(2);
        // Serial.print("Current Scoop motor : ");
        // Serial.print(currentS);
        pos_Scoop = checkHeight();
        // Serial.print(" ; Position Scoop motor : ");
        // Serial.println(pos_Scoop);
    }
    mS.write(noSpeedScoop);
    // Serial.println("ISEs measurement finished!");
    // delay(100);
    // Serial.println("Cleaning up the ISEs");
    // Serial.flush();
}
break;
}
}

if (COM == 'W') {
    currentW = 0;
    currentW = checkCurrent(1); //check current White motor 1
    currentS = checkCurrent(2); //check current Scoop Black motor 2
    dtostrf(currentW, 4, 1, Amp_W);
    dtostrf(currentS, 4, 1, Amp_S);
    sprintf(current_buffer, "%s ; %s", Amp_W, Amp_S); // sprintf(buffer tempat ngesave, format, buffer yang dibaca)
    Serial.print("Current: ");
    Serial.println(current_buffer);
    Serial.flush();
}
if (currentW > maxCurr || currentS > maxCurr) {
    Serial.println("High Current");
    mW.write(noSpeedWhite);
    mS.write(noSpeedScoop);
}
Serial.flush();
}

int checkCurrent(int Wmotor) {
    switch (Wmotor) {
        case 1:
            I_raw = analogRead(pinCSW); //white motor
            break;
        case 2:
            I_raw = analogRead(pinCSS); //scoop motor
            break;
    }
    I = (I_raw - 510) * 5 / 1024 / 0.04 - 0.04; // Data processing:510-raw data from analogRead when the input is 0;
    // 5-5v; the first 0.04-0.04V/A(sensitivity); the second 0.04-offset val; current in A
    // https://goo.gl/8LT7Rk
    delay(30);
    return abs(I);
}

int checkHeight() {
    pos_raw_Scoop = analogRead(pinSD);
    // TODO
    // height_Scoop = (0.0194 * pos_raw_Black - 0.5237) * 2.54; // in inch, convert to cm
    return pos_raw_Scoop;
}

```

9.2.3. OSA DAQ Code

```
//3 Sep 2018 for VI OSA PID OSA_colorSensor_EC_oldTentacle_VI

#include <Wire.h>
#include <stdio.h>
#include <OneWire.h>
#include <SPI.h>           //pin 11,12,13
#include <SD.h>
#include <SoftwareSerial.h>           //BT password : 1234
#define PMTK_SET_NMEA_OUTPUT_RMC "$PMTK314,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0*29"
#define PMTK_SET_NMEA_UPDATE_5HZ "$PMTK220,200*2C"
#define PMTK_API_SET_FIX_CTL_5HZ "$PMTK300,200,0,0,0,0*2F"
#define rx A2
#define tx A3

/* -- CONSTANTS --/
const int selectSD_pin = 10;
SoftwareSerial GPS_serial(8, 7);
SoftwareSerial BTserial(9, 6);           // TX | RX
SoftwareSerial Col_serial(rx, tx);     //define how the soft serial port is going to work
const int baud = 9600;
const int ledPinG = 5;                 //the number of the LED green pin
const int ledPinB = 4;                 //the number of the LED blue pin
const int ledPinR = 3;                 //the number of the LED red pin
const int ledPinRGB = 2;              //the number of the LED RGB pin, jangan pake A4 karena I2C pake pin A4 dan A5
http://playground.arduino.cc/Learning/Pins
const int buttonPin = A0;              //the number of button pin
const int thermistorPin = A1;         //thermistor temperature from EC probe https://goo.gl/OyCvHi
const unsigned int NMEA = 80;
static unsigned int input_pos = 0;
const int channel_id[] = {95, 96, 97, 98};
const int READ_BUFFER_SIZE = 80;
const int FLOAT_BUFFER_SIZE = 8;
const byte addr[8] = {40, 255, 51, 125, 2, 22, 3, 92};
//int debounce = 20;                   //debounce (from High to Low vice versa) time in ms
//int holdTime = 3000;                 //hold time when button pressed
File logfile;

/*--- Variables ---*/
//int buttonLast = 0;                  //buffered value of the button's previous state
//long buttonDownTime;                 //time button pressed down
//long buttonUpTime;                   //time button pressed up
//boolean ignoreUp = false;            //ignore button up when in press+hold event so the quick press+release event wont occur
String Smessage;
String Bmessage;
char SCOM;                             //char for serial read by arduino
char BCOM;                             //char for BT serial read by arduino
float Sum0, Sum1, Sum2, Sum3, Avg0, Avg1, Avg2, Avg3; // for tare
static char GPS_data[NMEA];
unsigned long current_time;
char channel_0[FLOAT_BUFFER_SIZE];
char channel_1[FLOAT_BUFFER_SIZE];
char channel_2[FLOAT_BUFFER_SIZE];
char channel_3[FLOAT_BUFFER_SIZE];
char read_buffer[READ_BUFFER_SIZE];
char temp_out[FLOAT_BUFFER_SIZE];
String Colstring = "";                  //a string to hold the data from the Atlas Scientific product
char inchar;
char *red;                             //char pointer used in string parsing
char *grn;                             //char pointer used in string parsing
char *blu;                             //char pointer used in string parsing
float int_red;                          //used to hold an int that is the color red
float int_grn;                          //used to hold an int that is the color green
float int_blu;                          //used to hold an int that is the color blue
char red_out[FLOAT_BUFFER_SIZE];
```

```

char grn_out[FLOAT_BUFFER_SIZE];
char blu_out[FLOAT_BUFFER_SIZE];
///https://goo.gl/SoHY3H
  // GPS_serial.begin(baud);       //otomatis nutup Col serial
  // //GPS
  // GPS_serial.println(F(PMTK_SET_NMEA_OUTPUT_RMC));
  // GPS_serial.println(F(PMTK_API_SET_FIX_CTL_5HZ));
  // GPS_serial.println(F(PMTK_SET_NMEA_UPDATE_5HZ));
  Wire.begin();
  //SD
  pinMode(selectSD_pin, OUTPUT);
  BTserial.begin(baud);           // open GPS serial port, otomatis nutup BTserial
  //initialize SD card and check VCC
  SDCard();
}

void loop() {
  // Keep reading from HC-05 and send to Arduino Serial Monitor
  // if (BTserial.available() > 0)
  if (BTserial.available() > 0 || Serial.available() > 0 )
  {
    Smessage = " ";
    Smessage = Serial.readString();
    Bmessage = " ";
    Bmessage = BTserial.readString();
    SCOM = Smessage.charAt(0);
    BCOM = Bmessage.charAt(0);
    // BTserial.println(COM);
    switch (BCOM) {
      case 'T':
        // BTserial.println(COM);
        ReadGPStentacle();
        break;
      case 'Z':
        // BTserial.println(COM);
        Tare();
        break;
    }
    switch (SCOM) {
      case 'T':
        // BTserial.println(COM);
        ReadGPStentacle();
        break;
      case 'Z':
        // BTserial.println(COM);
        Tare();
        break;
    }
  }
}

```

```

}
}
}

//initialize SD card and check VCC
void SDCard() {
  if (!SD.begin(selectSD_pin)) {
  }
  char filename[12];
  strcpy(filename, "LOG000.TXT"); //FAT 8.3 format only <=8 character
  for (uint16_t i = 0; i < 1000; i++) { //uint16_t between 0 to 65,535, before was uint8_t (0 to 255) jadi max file generated LOG255.TXT
    filename[3] = '0' + i / 100;
    filename[4] = '0' + (i / 10) % 10;
    filename[5] = '0' + i % 10;
    // create if does not exist
    if (!SD.exists(filename)) {
      break;
    }
  }
  logfile = SD.open(filename, FILE_WRITE);
  Serial.println(filename);
  BTserial.println(filename);
  logfile.println(filename);
  int VCC = readVcc();
  if (VCC > 4500 && logfile == true) {
    //system ready to log and voltage good so turn on LED green
    digitalWrite(ledPinG, HIGH);
    Serial.print(F("VCC : ")); BTserial.print(F("VCC : ")); logfile.print(F("VCC : "));
    Serial.println(VCC); BTserial.println(VCC); logfile.println(VCC);
  }
}

//check VCC
long readVcc() {
  // Read 1.1V reference against AVcc
  // set the reference to Vcc and the measurement to the internal 1.1V reference
  #if defined(__AVR_ATmega32U4__) || defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
    ADMUX = _BV(REFS0) | _BV(MUX4) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
  #elif defined(__AVR_ATtiny24__) || defined(__AVR_ATtiny44__) || defined(__AVR_ATtiny84__)
    ADMUX = _BV(MUX5) | _BV(MUX0);
  #elif defined(__AVR_ATtiny25__) || defined(__AVR_ATtiny45__) || defined(__AVR_ATtiny85__)
    ADMUX = _BV(MUX3) | _BV(MUX2);
  #else
    ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
  #endif
  delay(2); // Wait for Vref to settle
  ADCSRA |= _BV(ADSC); // Start conversion
  while (bit_is_set(ADCSRA, ADSC)); // measuring
  uint8_t low = ADCL; // must read ADCL first - it then locks ADCH
  uint8_t high = ADCH; // unlocks both
  long result = (high << 8) | low;
  result = 1125300L / result; // Calculate Vcc (in mV); 1125300 = 1.1*1023*1000
  return result; // Vcc in millivolts
}

void Tare() {
  digitalWrite(ledPinG, LOW);
  digitalWrite(ledPinRGB, HIGH);
  int i = 1;
  Sum0 = 0; Sum1 = 0; Sum2 = 0; Sum3 = 0; Avg0 = 0; Avg1 = 0; Avg2 = 0; Avg3 = 0;
  for (i; i <= 15; i++) {
    // retrieve the data
    float ch0_T = retrieve_channel(95); // _R means READ
    Sum0 = Sum0 + ch0_T;
    delay(10);
  }
}

```

```

// send command to read
read_channel(95);
delay(250);
float ch1_T = retrieve_channel(96);
Sum1 = Sum1 + ch1_T;
delay(10);
read_channel(96);
delay(250);
float ch2_T = retrieve_channel(97);
Sum2 = Sum2 + ch2_T;
delay(10);
read_channel(97);
delay(250);
float ch3_T = retrieve_channel(98);
Sum3 = Sum3 + ch3_T;
delay(10);
read_channel(98);
delay(250);
}
Avg0 = Sum0 / 14.0; Avg1 = Sum1 / 14.0; Avg2 = Sum2 / 14.0; Avg3 = Sum3 / 14.0; //dibagi 14 soalnya yg pertama 0, belum ngomand "r"
Serial.print(F("Avg0 : ")); BTserial.print(F("Avg0 : ")); logfile.print(F("Avg0 : ")); Serial.println(Avg0); BTserial.println(Avg0); logfile.println(Avg0);
Serial.print(F("Avg1 : ")); BTserial.print(F("Avg1 : ")); logfile.print(F("Avg1 : ")); Serial.println(Avg1); BTserial.println(Avg1); logfile.println(Avg1);
Serial.print(F("Avg2 : ")); BTserial.print(F("Avg2 : ")); logfile.print(F("Avg2 : ")); Serial.println(Avg2); BTserial.println(Avg2); logfile.println(Avg2);
Serial.print(F("Avg3 : ")); BTserial.print(F("Avg3 : ")); logfile.print(F("Avg3 : ")); Serial.println(Avg3); BTserial.println(Avg3); logfile.println(Avg3);
Serial.println(F("")); BTserial.println(F("")); logfile.println(F(""));
logfile.close();
Serial.flush();
digitalWrite(ledPinRGB, LOW);
SDCard();
}

//read GPS and Tentacle main function
void ReadGPStentacle() {
long int TIME = millis() + 3000; //time for GPS reading
long int TIME2 = millis() + 30000; //time for Tentacle reading
// open GPS serial port, otomatis nutup BTserial
GPS_serial.begin(baud);
//GPS print setup
GPS_serial.println(F(PMTK_SET_NMEA_OUTPUT_RMC));
GPS_serial.println(F(PMTK_API_SET_FIX_CTL_5HZ));
GPS_serial.println(F(PMTK_SET_NMEA_UPDATE_5HZ));
Serial.println(F("GPS data :")); BTserial.println(F("GPS data :")); logfile.println(F("GPS data :"));
digitalWrite(ledPinG, LOW); //turn off green LED
while (millis() < TIME) {
GPSread();
}
// open BT serial port, otomatis nutup GPS serial. buka BT serial untuk COM BTserial.available di loop
BTserial.begin(baud);
//read Tentacle
Serial.println(F("Tentacle data :")); Serial.println(F("Time; ch0; ch1; ch2; ch3; Red; Green; Blue; Temp; Checksum"));
BTserial.println(F("Tentacle data :")); BTserial.println(F("Time; ch0; ch1; ch2; ch3; Red; Green; Blue; Temp; Checksum"));
logfile.println(F("Tentacle data :")); logfile.println(F("Time; ch0; ch1; ch2; ch3; Red; Green; Blue; Temp; Checksum"));
digitalWrite(ledPinR, HIGH); //turn on red LED
while (millis() < TIME2) {
TENTACLEread();
}
logfile.close();
Serial.flush();
BTserial.flush();
digitalWrite(ledPinR, LOW); //turn off red LED
BTserial.begin(baud);
SDCard();
}

//read GPS function
void GPSread() {

```

```

if (GPS_serial.available() > 0) {
  const byte c = GPS_serial.read();
  switch (c)
  {
    default:
      if (input_pos < (NMEA - 1))
        GPS_data[input_pos++] = c;
      break;
    case '\r':
      //discard carriage return
      break;
    case '\n':
      //detect end of line and print the complete sentence
      GPS_data [input_pos] = 0; //biasanya null byte at the end of sentence tapi untuk NMEA sentence gak pake null byte tapi pake
      //CRLF jadi line ini gak usah dipake http://www.gpsinformation.org/dale/nmea.htm
      char*GPS_in = GPS_data;
      char fix = GPS_data[18];
      if (fix == 'A') { //https://www.arduino.cc/en/Tutorial/StringCharacters
        digitalWrite(ledPinB, HIGH); //turn on blue LED
      }
      Serial.println(GPS_in); BTserial.println(GPS_in); logfile.println(GPS_in);
      input_pos = 0;
      break;
  }
  digitalWrite(ledPinB, LOW); //turn off blue LED
}
}

//read Tentacle function
void TENTACLEread() {
  //prepare to read the temperature sensor here, because this sensor need to wait (delay, 750ms ideally) for ADC conversion to be completed.
  //so we eliminate the delay by putting here although it is not 750ms but 400ms (4x100ms) from tentacle read
  //read tentacle+millis
  //read color+millis+tentacle
  //retrieve the data
  Col_serial.begin(baud);
  Col_serial.print('R');
  Col_serial.print('\r');
  while (Col_serial.available() > 0) {
    inchar = Col_serial.read(); //get the char we just received
    Colstring += inchar; //add the char to the var called Colstring
    if (inchar == '\r' && Colstring.indexOf("**") == -1) { //if the incoming character is a <CR> and https://goo.gl/JizLYK
      char Colstring_array[12]; //we make a char array
      Colstring.toCharArray(Colstring_array, 12); //convert the string to a char array
      red = strtok(Colstring_array, ","); //let's pars the array at each comma
      grn = strtok(NULL, ","); //let's pars the array at each comma
      blu = strtok(NULL, ","); //let's pars the array at each comma
      int_red = atof(red);
      int_grn = atof(grn);
      int_blu = atof(blu);
    }
  }
  Colstring = ""; //clear the string
  float ch0_R = retrieve_channel(95); // _R means READ
  float ch0_N = ch0_R - Avg0; // _N means after Nulling
  delay(10);
  // send command to read
  read_channel(95);
  delay(250);
  float ch1_R = retrieve_channel(96);
  float ch1_N = ch1_R - Avg1;
  delay(10);
  read_channel(96);
  delay(250);
  float ch2_R = retrieve_channel(97);
  float ch2_N = ch2_R - Avg2;
  delay(10);
  read_channel(97);
}

```

```

delay(250);
float ch3_R = retrieve_channel(98);
float ch3_N = ch3_R - Avg3;
delay(10);
read_channel(98);
delay(250);
current_time = millis();
float temp_raw = analogRead(thermistorPin);
float temp_mV = temp_raw * (5.0 / 1023.0) * 1000;
float temp_C = temp_mV * 1;
//Functions to read and store data from each channel
dtostrf(ch0_N, 6, 1, channel_0);
dtostrf(ch1_N, 6, 1, channel_1);
dtostrf(ch2_N, 6, 1, channel_2);
dtostrf(ch3_N, 6, 1, channel_3);
dtostrf(temp_C, 4, 1, temp_out);
dtostrf(int_red, 3, 0, red_out);
dtostrf(int_grn, 3, 0, grn_out);
dtostrf(int_blu, 3, 0, blu_out);
sprintf(read_buffer, "%li;%s;%s;%s;%s;%s;%s;%s", current_time, channel_0, channel_1, channel_2, channel_3, red_out, grn_out, blu_out,
temp_out);
//call checksum function
int chksum = checksum(read_buffer);
Serial.print(read_buffer); Serial.print(F("; chk: ")); Serial.println(chksum);
logfile.print(read_buffer); logfile.print(F("; chk: ")); logfile.println(chksum);
BTserial.print(read_buffer); BTserial.print(F("; chk: ")); BTserial.println(chksum);
}

void read_channel(int channel_id) {
Wire.beginTransmission(channel_id); //Begin a transmission to the I2C slave device with the given address i.e. 95
Wire.write('r'); //master write
Wire.endTransmission(); //end transmission
}

float retrieve_channel(int channel_id) {
char sensordata[8]; //reserved 8 bytes buffer size for Wire library
byte sensor_bytes_received = 0; //We need to know how many characters bytes have been received
byte read_byte = 0; //read bytes one by one from Slave; output is an unsigned integer
sensor_bytes_received = 0; //reset data counter;
memset(sensordata, 0, sizeof(sensordata)); //clear sensordata array;
Wire.requestFrom(channel_id, 32); //Used by the master to request bytes from a slave device.
Wire.read(); //read first byte from slave (command 'r' success?); must have read here
while (Wire.available ()) { //Returns the number of bytes available for retrieval with read()
read_byte = Wire.read(); //Reads a byte that was transmitted from a slave device
if (read_byte == 0) { //null character indicates 0 byte information or end of reading
Wire.endTransmission(); //end the I2C data transmission.
break; //exit the while loop
}
else {
sensordata[sensor_bytes_received] = read_byte; //add this byte to the sensor data array.
sensor_bytes_received++;
}
}
return atof (sensordata); //return anything to float (atof)
}

//checksum function
int checksum(char* buf) {
int sum = 0;
for (int i = 0 ; i < READ_BUFFER_SIZE; i++) {
sum = sum + buf[i];
}
return sum % 256;
}

```

9.3. PD Compensation Matlab Code

Poles of closed loop G(s) :

```
numg=327.7;           % numerator of G(s).
deng=[1 16.51 327.7]; % denominator of G(s).
G=tf(numg,deng)      % Create G(s).
T=feedback(G,1);    % Form equivalent closed-loop tf
poles=pole(T)       % Display closed-loop poles.
```

%% Reference : (Nise, 2015)

```
clear           % Clear variables from workspace.
clf            % Clear graph.
%%%%%%%%%%%%% Input and create the uncompensated system %%%%%%%%%%%%%%
numg=327.7;    % Numerator of G(s).
deng=[1 16.51 327.7]; % Denominator of G(s).
G=tf(numg,deng); % G(s).
rlocus(G)     % Show root locus
pos=20;       % %Overshoot
zeta=0.46;    % Zeta
sgrid(zeta,0) % Overlay rootlocus with damping ratio
title(['Uncompensated Root Locus with ', num2str(pos),'% Overshoot Line'])
%%%%%%%%%%%%% Closed-loop poles(check stability) %%%%%%%%%%%%%%
T=feedback(G,1); % Find closed loop equivalent of G(s)
p=pole(T);      % Closed-loop poles.
%%%%%%%%%%%%% Calculate simulation's uncompensated system specifications %%%%%%%%%%%%%%
[K,p]=rlocfind(G); % Generate gain, K, and closed-loop poles, p
p      % Display uncompensated closed-loop poles.
f = input('Give pole number that is operating point:');
p_dom=p(f);    % Uncompensated dominant pole.
p_real = real(p_dom); % Extract real part from dominant pole
p_imag = imag(p_dom); % Extract imaginary part of dominant pole
gain=K         % Display uncompensated Gain
z=zeta        % Display uncompensated Zeta
OS = pos      % Display %OS
Ts = 4/abs(p_real) % Calculate Settling Time
Wn=4/(Ts*zeta) % Calculate natural frequency
Tp = pi/abs(p_imag) % Calculate Peak Time
WnTr = 1.76*zeta^3-0.417*zeta^2+1.039*zeta+1;
Tr = WnTr/Wn   % calculate rise time
%%%%%%%%%%%%% Evaluate tf (find steady state error) n=0 -> type 0 %%%%%%%%%%%%%%
Kp=dcgain(G)*K % Evaluate Kp=numg/deng for s=0.
ess=1/(1+Kp)   % Uncompensated steady-state error
%%%%%%%%%%%%% Step input to see response of the whole uncompensated system %%%%%%%%%%%%%%
T=feedback(K*G,1); % Find uncompensated T(s).
step(T)        % Plot step response
title(['Uncompensated System Step Response with ', num2str(pos),'% Overshoot']);
%%%%%%%%%%%%% PD controller design with improved settling time %%%%%%%%%%%%%%
'Press any key to go to PD compensation'
pause
Tsc=input('Type Desired Settling Time: ');
Wnc=4/(Tsc*zeta); % Calculate natural frequency
pc=(-zeta*Wnc)+(Wnc*sqrt(1-zeta^2)*i); % Calculate pole location from desired Ts
s = pc;          % Evaluate s at desired pole
poly_G = 1/(s^2+16.51*s+327.7);
Theta = (180/pi)*(angle(poly_G)); % Sum of angle at desired pole pc
PD_angle=abs(180-Theta); % Calculate zero angle
zc=-((imag(pc)/tan(PD_angle*pi/180))+real(pc)); % Calculate PD zero location.
%%%%%%%%%%%%% PD controller transfer function %%%%%%%%%%%%%%
numc=[1 -zc]; % Calculate numerator of Gc(s).
denc=[0 1]; % Calculate denominator of Gc(s).
Gc=tf(numc,denc); % Create PD controller transfer function
Ge=Gc*G; % Cascade G(s)*Gc(s)
rlocus(Ge) % Plot root locus of PD compensated system.
sgrid(zeta,0) % Overlay with zeta
title(['PD Compensated Root Locus with ', num2str(pos),'%Overshoot Line'])
%%%%%%%% To zoom rlocus graph %%%%%%%%%
x1 = input('Boundary zoom in x1:');
```

```

x2 = input('Boundary zoom in x2:');
y1 = input('Boundary zoom in y1:');
y2 = input('Boundary zoom in y2:');
axis([x1 x2 y1 y2])           % Range close-up view.
[Kpd,ppd]=rlocfind(Ge);       % Generate gain, K, and closed-loop poles, p
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Calculate simulation's PD compensated system specifications %%%%%%%%%%%%%%%
ppd                            % Display PD compensated closed-loop poles.
fpd=input('Give pole number that is operating point:');
gainpd=Kpd                     % Display PD compensated gain.
z=zeta                          % Display PD compensated Zeta.
OS = pos                        % Display %OS
ppd_dom=ppd(fpd);              % PD compensated dominant pole.
ppd_real = real(ppd_dom);       % Extract real part from dominant pole
ppd_imag = imag(ppd_dom);       % Extract imaginary part of dominant pole
Tspd = 4/abs(ppd_real)          % Calculate Settling Time
Tppd = pi/abs(ppd_imag)         % Calculate Peak Time
WnTrpd = 1.76*zeta^3-0.417*zeta^2+1.039*zeta+1;
Wnppd = sqrt(ppd_real^2+ppd_imag^2) % Display PD compensated natural freq.
Trpd = WnTrpd/Wnppd            % Calculate rise time
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Evaluate tf (find steady state error) n=0 -> type 0 %%%%%%%%%%%%%%%
Kp_pd_ss=dcgain(Ge)*Kpd        % Evaluate Kp=numg/deng for s=0.
ess_pd=1/(1+Kp_pd_ss)          % PD compensated steady-state error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Step input to see response of the whole PD compensated system %%%%%%%%%%%%%%%
Tpd=feedback(Kpd*Ge,1);        % Create PD compensated T(s).
step(Tpd);                     % Plot step response PD compensated system.
hold on
step(T)                         % Plot step response of uncompensated system.
title(['PD Compensated and Uncompensated Closed Loop Step Response']);
hold off

```

9.4. Manual Soil Sampler DAQ Code

```
//28 Mar 2018 combine 4ch BT temp old circuit 1 seconds remap pin combine_v24_4ch_BT_temp_old_circuit_1s_noY360
#include <Wire.h>
#include <stdio.h>
#include <OneWire.h>
#include <SPI.h>           //pin 11,12,13
#include <SD.h>
#include <SoftwareSerial.h>           //BT password : 1234
#define PMTK_SET_NMEA_OUTPUT_RMC "$PMTK314,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0*29"
#define PMTK_SET_NMEA_UPDATE_5HZ "$PMTK220,200*2C"
#define PMTK_API_SET_FIX_CTL_5HZ "$PMTK300,200,0,0,0*2F"
/* -- CONSTANTS --*/
const int selectSD_pin = 10;
SoftwareSerial GPS_serial(8, 7);
SoftwareSerial BTserial(9, 6);           // TX | RX
const int ledPinG = 5;           //the number of the LED green pin
const int ledPinB = 4;           //the number of the LED blue pin
const int ledPinR = 3;           //the number of the LED red pin
const int ledPinRGB = 2;           //the number of the LED RGB pin, jangan pake A4 karena I2C pake pin A4 dan A5
http://playground.arduino.cc/Learning/Pins
const int buttonPin = A0;           //the number of button pin
const int baud = 9600;
const unsigned int NMEA = 80;
static unsigned int input_pos = 0;
const int channel_id[] = {95, 96, 97, 98};
const int READ_BUFFER_SIZE = 45;
const int FLOAT_BUFFER_SIZE = 8;
OneWire tempPin(A2);           //pin A2
const byte addr[8] = {40, 255, 51, 125, 2, 22, 3, 92};
int debounce = 20;           //debounce (from High to Low vice versa) time in ms
int holdTime = 3000;           //hold time when button pressed
File logfile;
/*--- Variables ---*/
int buttonLast = 0;           //buffered value of the button's previous state
long buttonDownTime;           //time button pressed down
long buttonUpTime;           //time button pressed up
boolean ignoreUp = false;           //ignore button up when in press+hold event so the quick press+release event wont occur
float Sum0, Sum1, Sum2, Sum3, Avg0, Avg1, Avg2, Avg3; // for tare
static char GPS_data[NMEA];
unsigned long current_time;
char channel_0[FLOAT_BUFFER_SIZE];
char channel_1[FLOAT_BUFFER_SIZE];
char channel_2[FLOAT_BUFFER_SIZE];
char channel_3[FLOAT_BUFFER_SIZE];
char read_buffer[READ_BUFFER_SIZE];
char t_C[FLOAT_BUFFER_SIZE];
byte data[10];           //temp byte data
byte tempPresent;
float celsius;

void setup() {
  //LED pin
  pinMode(ledPinG, OUTPUT);
  pinMode(ledPinB, OUTPUT);
  pinMode(ledPinR, OUTPUT);
  pinMode(ledPinRGB, OUTPUT);
  //Button pin
  pinMode(buttonPin, INPUT_PULLUP);
  //BAUD
  Serial.begin (baud);
  BTserial.begin(baud);
  GPS_serial.begin(baud);
  Wire.begin();
  //GPS
  GPS_serial.println(F(PMTK_SET_NMEA_OUTPUT_RMC));
```

```

GPS_serial.println(F(PMTK_API_SET_FIX_CTL_5HZ));
GPS_serial.println(F(PMTK_SET_NMEA_UPDATE_5HZ));
//SD
pinMode(selectSD_pin, OUTPUT);
//initialize SD card and check VCC
SDCard();
buttonLast = HIGH;           //supaya awalnya 1 1, nggak 1 0
}

void loop() {
int buttonState = digitalRead(buttonPin);
if (buttonState == LOW && buttonLast == HIGH && (millis() - buttonUpTime) > long(debounce))
{
buttonDownTime = millis();
}
if (buttonState == HIGH && buttonLast == LOW && (millis() - buttonDownTime) > long(debounce))
{
if (ignoreUp == false) {
ReadGPStentacle();
}
else {
ignoreUp = false;
}
buttonUpTime = millis();
}
if (buttonState == LOW && (millis() - buttonDownTime) > long(holdTime))
{
Tare();
ignoreUp = true;
buttonDownTime = millis();
}
buttonLast = buttonState;
}

//initialize SD card and check VCC
void SDCard() {
if (!SD.begin(selectSD_pin)) {
}
char filename[12];
strcpy(filename, "LOG000.TXT");           //FAT 8.3 format only <=8 character
for (uint16_t i = 0; i < 1000; i++) {     //uint16_t between 0 to 65,535, before was uint8_t (0 to 255) jadi max file generated LOG255.TXT
filename[3] = '0' + i / 100;
filename[4] = '0' + (i / 10) % 10;
filename[5] = '0' + i % 10;
// create if does not exist
if (!SD.exists(filename)) {
break;
}
}
logfile = SD.open(filename, FILE_WRITE);
Serial.println(filename);
BTserial.println(filename);
int VCC = readVcc();
if (VCC > 4500 && logfile == true && tempPresent == 1) {
//system ready to log and voltage good so turn on LED green
digitalWrite(ledPinG, HIGH);
Serial.print(F("VCC : ")); BTserial.print(F("VCC : ")); logfile.print(F("VCC : "));
Serial.println(VCC); BTserial.println(VCC); logfile.println(VCC);
}
}
//check VCC
long readVcc() {
// Read 1.1V reference against AVcc
// set the reference to Vcc and the measurement to the internal 1.1V reference
#ifdef __AVR_ATmega32U4__ || defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
ADMUX = _BV(REFS0) | _BV(MUX4) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);

```

```

#elif defined (__AVR_ATtiny24__) || defined(__AVR_ATtiny44__) || defined(__AVR_ATtiny84__)
  ADMUX = _BV(MUX5) | _BV(MUX0);
#elif defined (__AVR_ATtiny25__) || defined(__AVR_ATtiny45__) || defined(__AVR_ATtiny85__)
  ADMUX = _BV(MUX3) | _BV(MUX2);
#else
  ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
#endif
delay(2); // Wait for Vref to settle
ADCSRA |= _BV(ADSC); // Start conversion
while (bit_is_set(ADCSRA, ADSC)); // measuring
uint8_t low = ADCL; // must read ADCL first - it then locks ADCH
uint8_t high = ADCH; // unlocks both
long result = (high << 8) | low;
result = 1125300L / result; // Calculate Vcc (in mV); 1125300 = 1.1*1023*1000
return result; // Vcc in millivolts
}
void Tare() {
  digitalWrite(ledPinG, LOW);
  digitalWrite(ledPinRGB, HIGH);
  int i = 1;
  Sum0 = 0; Sum1 = 0; Sum2 = 0; Sum3 = 0; Avg0 = 0; Avg1 = 0; Avg2 = 0; Avg3 = 0;
  for (i; i <= 15; i++) {
    // retrieve the data
    float ch0_T = retrieve_channel(95); // _R means READ
    Sum0 = Sum0 + ch0_T;
    delay(10);
    // send command to read
    read_channel(95);
    delay(250);
    float ch1_T = retrieve_channel(96);
    Sum1 = Sum1 + ch1_T;
    delay(10);
    read_channel(96);
    delay(250);
    float ch2_T = retrieve_channel(97);
    Sum2 = Sum2 + ch2_T;
    delay(10);
    read_channel(97);
    delay(250);
    float ch3_T = retrieve_channel(98);
    Sum3 = Sum3 + ch3_T;
    delay(10);
    read_channel(98);
    delay(250);
  }
  Avg0 = Sum0 / 14.0; Avg1 = Sum1 / 14.0; Avg2 = Sum2 / 14.0; Avg3 = Sum3 / 14.0; //dibagi 14 soalnya yg pertama 0, belum ngomand "r"
  Serial.print(F("Avg0 : ")); BTserial.print(F("Avg0 : ")); logfile.print(F("Avg0 : ")); Serial.println(Avg0); BTserial.println(Avg0); logfile.println(Avg0);
  Serial.print(F("Avg1 : ")); BTserial.print(F("Avg1 : ")); logfile.print(F("Avg1 : ")); Serial.println(Avg1); BTserial.println(Avg1); logfile.println(Avg1);
  Serial.print(F("Avg2 : ")); BTserial.print(F("Avg2 : ")); logfile.print(F("Avg2 : ")); Serial.println(Avg2); BTserial.println(Avg2); logfile.println(Avg2);
  Serial.print(F("Avg3 : ")); BTserial.print(F("Avg3 : ")); logfile.print(F("Avg3 : ")); Serial.println(Avg3); BTserial.println(Avg3); logfile.println(Avg3);
  Serial.println(F("")); BTserial.println(F("")); logfile.println(F(""));
  logfile.close();
  Serial.flush();
  digitalWrite(ledPinRGB, LOW);
  SDCard();
}
//read GPS and Tentacle main function
void ReadGPSstentacle() {
  long int TIME = millis() + 5000; //time for GPS reading
  long int TIME2 = millis() + 90000; //time for Tentacle reading
  //read GPS
  Serial.println(F("GPS data :")); BTserial.println(F("GPS data :")); logfile.println(F("GPS data :"));
  digitalWrite(ledPinG, LOW); //turn off green LED
  while (millis() < TIME) {
    GPSread();
  }
}

```

```

//read Tentacle
Serial.println(F("Tentacle data :")); Serial.println(F("Time; ch0; ch1; ch2; ch3; Temp;Checksum"));
BTserial.println(F("Tentacle data :")); BTserial.println(F("Time; ch0; ch1; ch2; ch3; Temp;Checksum"));
logfile.println(F("Tentacle data :")); logfile.println(F("Time; ch0; ch1; ch2; ch3; Temp;Checksum"));
digitalWrite(ledPinR, HIGH); //turn on red LED
while (millis() < TIME2) {
  TENTACLEread();
}
logfile.close();
Serial.flush();
digitalWrite(ledPinR, LOW); //turn off red LED
SDCard();
}
//read GPS function
void GPSread() {
if (GPS_serial.available() > 0) {
  const byte c = GPS_serial.read();
  switch (c)
  {
  default:
    if (input_pos < (NMEA - 1))
      GPS_data[input_pos++] = c;
    break;
  case '\r': //discard carriage return
    break;
  case '\n': //detect end of line and print the complete sentence
    GPS_data[input_pos] = 0; //biasanya null byte at the end of sentence tapi untuk NMEA sentence gak pake null byte tapi pake
    CRLF jadi line ini gak usah dipake http://www.gpsinformation.org/dale/nmea.htm
    char*GPS_in = GPS_data;
    char fix = GPS_data[18];
    if (fix == 'A') { //https://www.arduino.cc/en/Tutorial/StringCharacters
      digitalWrite(ledPinB, HIGH); //turn on blue LED
    }
    Serial.println(GPS_in); BTserial.println(GPS_in); logfile.println(GPS_in);
    input_pos = 0;
    break;
  }
  digitalWrite(ledPinB, LOW); //turn off blue LED
}
}
//read Tentacle function
void TENTACLEread() {
//prepare to read the temperature sensor here, because this sensor need to wait (delay, 750ms ideally) for ADC conversion to be completed.
//so we eliminate the delay by putting here although it is not 750ms but 400ms (4x100ms) from tentacle read
tempPin.reset();
tempPin.select(addr);
tempPin.write(0x44, 1);
//read tentacle+millis
// retrieve the data
float ch0_R = retrieve_channel(95); //_R means READ
float ch0_N = ch0_R - Avg0; //_N means after Nulling
delay(10);
// send command to read
read_channel(95);
delay(250);
float ch1_R = retrieve_channel(96);
float ch1_N = ch1_R - Avg1;
delay(10);
read_channel(96);
delay(250);
float ch2_R = retrieve_channel(97);
float ch2_N = ch2_R - Avg2;
delay(10);
read_channel(97);
delay(250);
float ch3_R = retrieve_channel(98);

```

```

float ch3_N = ch3_R - Avg3;
delay(10);
read_channel(98);
delay(250);
current_time = millis();
float temp_C = tempRead();
//Functions to read and store data from each channel
dtostrf(ch0_N, 6, 1, channel_0);
dtostrf(ch1_N, 6, 1, channel_1);
dtostrf(ch2_N, 6, 1, channel_2);
dtostrf(ch3_N, 6, 1, channel_3);
dtostrf(temp_C, 5, 1, t_C);
sprintf(read_buffer, "%li;%s;%s;%s;%s", current_time, channel_0, channel_1, channel_2, channel_3, t_C);
// //call checksum function
int chksum = checksum(read_buffer);
Serial.print(read_buffer); Serial.print(F("; chk: ")); Serial.println(chksum);
logfile.print(read_buffer); logfile.print(F("; chk: ")); logfile.println(chksum);
BTserial.print(read_buffer); BTserial.print(F("; chk: ")); BTserial.println(chksum);
}

void read_channel(int channel_id) {
Wire.beginTransmission(channel_id); //Begin a transmission to the I2C slave device with the given address i.e. 95
Wire.write('r'); //master write
Wire.endTransmission(); //end transmission
}

float retrieve_channel(int channel_id) {
char sensordata[8]; //reserved 8 bytes buffer size for Wire library
byte sensor_bytes_received = 0; //We need to know how many characters bytes have been received
byte read_byte = 0; //read bytes one by one from Slave; output is an unsigned integer
sensor_bytes_received = 0; //reset data counter;
memset(sensordata, 0, sizeof(sensordata)); //clear sensordata array;
Wire.requestFrom(channel_id, 32); //Used by the master to request bytes from a slave device.
Wire.read(); //read first byte from slave (command 'r' success?); must have read here
while (Wire.available ()) { //Returns the number of bytes available for retrieval with read()
read_byte = Wire.read(); //Reads a byte that was transmitted from a slave device
if (read_byte == 0) { //null character indicates 0 byte information or end of reading
Wire.endTransmission(); //end the I2C data transmission.
break; //exit the while loop
}
else {
sensordata[sensor_bytes_received] = read_byte; //add this byte to the sensor data array.
sensor_bytes_received++;
}
}
return atof (sensordata); //return anything to float (atof)
}

//read temperature
float tempRead() {
tempPresent = tempPin.reset(); //reset again
tempPin.select(addr); //select again
tempPin.write(0xBE); //Read the entire Scratchpad
for (int i = 0; i < 9; i++) { //we need 9 bytes
data[i] = tempPin.read(); //Read a byte from the Scratchpad
}
//convert ADC value to celcius
int16_t raw = (data[1] << 8) | data[0];
celsius = (float)raw / 16.0;
return celsius;
}

//checksum function
int checksum(char* buf) {
int sum = 0;
for (int i = 0 ; i < READ_BUFFER_SIZE; i++) {
sum = sum + buf[i];
}
return sum % 256
}

```

9.5. DUALEM-21S Default NMEA 0183 Data Stream

Block	Sentence	
1	<i>\$PDLMI,hhmmss.sss,wddd.d,xxxx.ee,ypppp.p,zqqq.qq*cc</i>	
	Sentence part	Description
	l	nominal array-length in m
	hhmmss.sss	hour-minute-second time of the measurement
	w, x, y, z	signs (e.g. + or -) of the measurement components
	ddd.d	HCP conductivity in mS.m ⁻¹
	eee.ee	HCP in phase in ppt
	pppp.p	PRP conductivity in mS.m ⁻¹
	qqq.qq	PRP in phase in ppt
	cc	hexadecimal checksum of values in the output
2	<i>\$PDLMA,ww.vv,xtt.t,ypp.p,zrr.r*cc</i>	
	Sentence part	Description
	w, x, y, z	signs (e.g. + or -) of the measurement components
	vv.vv	voltage applied to the sensor (V)
	tt.t	internal temperature of the sensor (°C)
	pp.p	pitch of the sensor (°)
	rr.r	roll of the sensor (°)
	cc	hexadecimal checksum of values in the output
3	<i>\$GPGGA,hhmmss,llnn.nnnn,d,ooopp.pppp,e,q,tt,uu.u,aaaa.a,M,gggg.g,M,ww,rrrr*cc</i>	
	Sentence part	Description
	hhmmss	hour-minute-second of coordinated universal time (UTC)
	ll	degrees of latitude
	nn.nnnn	minutes of latitude
	d	hemisphere of the latitude (e.g. N or S)
	ooo	degrees of longitude
	pp.pppp	minutes of longitude
	e	hemisphere of the longitude (e.g. E or W)
	q	quality of the GPS position
	tt	number of the satellites contributing to the GPS position
	uu.u	GPS horizontal dilution of precision
	aaaa.a	altitude above mean-sea-level
	M	units of height (m)
	gggg.g	geoidal height
	ww	number seconds since the last differential-GPS update
	rrrr	identification of the differential-GPS reference
cc	hexadecimal checksum of values in the NMEA-GPGGA sentence	

9.6. DUALEM-21S Filtering Code

```
function varargout = Dualem_Filtering_GUI(varargin)
% DUALEM_FILTERING_GUI MATLAB code for Dualem_Filtering_GUI.fig
%   DUALEM_FILTERING_GUI, by itself, creates a new DUALEM_FILTERING_GUI or raises the existing
%   singleton*.
%
%   H = DUALEM_FILTERING_GUI returns the handle to a new DUALEM_FILTERING_GUI or the handle to
%   the existing singleton*.
%
%   DUALEM_FILTERING_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in DUALEM_FILTERING_GUI.M with the given input arguments.
%
%   DUALEM_FILTERING_GUI('Property','Value',...) creates a new DUALEM_FILTERING_GUI or raises
the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before Dualem_Filtering_GUI_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to Dualem_Filtering_GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Dualem_Filtering_GUI

% Last Modified by GUIDE v2.5 31-Jul-2019 21:05:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Dualem_Filtering_GUI_OpeningFcn, ...
                  'gui_OutputFcn',  @Dualem_Filtering_GUI_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Dualem_Filtering_GUI is made visible.
function Dualem_Filtering_GUI_OpeningFcn(hObject,~, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Dualem_Filtering_GUI (see VARARGIN)

% Choose default command line output for Dualem_Filtering_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Dualem_Filtering_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```

% --- Outputs from this function are returned to the command line.
function varargout = Dualem_Filtering_GUI_OutputFcn(~, ~, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in dual_path_but.
function dual_path_but_Callback(~, ~, handles)
% hObject handle to dual_path_but (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
[dual_filename, dual_pathname] = uigetfile({'*.csv'}, 'Dualem data');
dual_fullpathname = strcat(dual_pathname, dual_filename);
dual_file = fileread(dual_fullpathname);
set(handles.dual_path_text2, 'String', dual_fullpathname); %show fullpathname
set(handles.dual_path_text3, 'String', dual_file); %show the data
dual_dat_read = readtable(dual_filename, 'Delimiter', ',', 'ReadVariableNames', true);
% dual_dat_filename = dual_fullpathname;
%guidata(hObject, handles);
assignin('base', 'dual_file', dual_file);
assignin('base', 'dual_dat_read', dual_dat_read);
assignin('base', 'pathname', dual_pathname);

% --- Executes on button press in start_filter.
function start_filter_Callback(~, ~, handles)
% hObject handle to start_filter (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% -----%
dual_dat_read = evalin('base', 'dual_dat_read'); %https://goo.gl/aE6w2Z
% add UTM lat lon
[x_utm, y_utm] = deg2utm(dual_dat_read.Latitude(:), dual_dat_read.Longitude(:));
% normalize UTM so we can see the distance and field size in m
x_utm_min = min(x_utm);
y_utm_min = min(y_utm);
%add 10 m buffer from minimum for better visualization
x_utm_norm = (x_utm - x_utm_min + 10);
y_utm_norm = (y_utm - y_utm_min + 10);
%convert to table, append to dual dat read, change name to dual_dat_r --> raw
dual_dat_raw = [array2table(x_utm_norm) array2table(y_utm_norm) dual_dat_read];
dual_dat_raw.Properties.VariableNames{'x_utm_norm'} = 'Long_UTM';
dual_dat_raw.Properties.VariableNames{'y_utm_norm'} = 'Lat_UTM';
figure;
scatter(dual_dat_raw.Long_UTM, dual_dat_raw.Lat_UTM, 4, 'm', 'o');
title(['DUALEM 21-S Raw Data (total: ' num2str(height(dual_dat_raw)) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_raw', dual_dat_raw);
% -----%
% check and delete data manually (selectdata function); press x to exit
dual_del = dual_dat_raw;
figure;
subplot(2, 2, 1);
scatter(dual_del.Long_UTM, dual_del.Lat_UTM, 4, dual_del.HCP10);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('Select HCP10 to be Deleted (press X to exit)');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2, 2, 2);
scatter(dual_del.Long_UTM, dual_del.Lat_UTM, 4, dual_del.PRP11);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';

```

```

title('Select PRP11 to be Deleted (press X to exit)');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,3);
scatter(dual_del.Long_UTM,dual_del.Lat_UTM,4,dual_del.HCP20);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('Select HCP20 to be Deleted (press X to exit)');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,4);
scatter(dual_del.Long_UTM,dual_del.Lat_UTM,4,dual_del.PRP21);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('Select PRP21 to be Deleted (press X to exit)');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
while true
    w = waitforbuttonpress;
    switch w
        case 1 %(keyboard press)
            key = get(gcf,'currentcharacter');
            switch key
                case 120 % 120 ASCII for lowercase x
                    close();
                    break;
            end
        end
        pl = selectdata('selectionmode','closest','action','delete');
        dual_del(pl,:) = [];
subplot(2,2,1);
scatter(dual_del.Long_UTM,dual_del.Lat_UTM,4,dual_del.HCP10);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('Select HCP10 to be Deleted (press X to exit)');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,2);
scatter(dual_del.Long_UTM,dual_del.Lat_UTM,4,dual_del.PRP11);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('Select PRP11 to be Deleted (press X to exit)');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,3);
scatter(dual_del.Long_UTM,dual_del.Lat_UTM,4,dual_del.HCP20);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('Select HCP20 to be Deleted (press X to exit)');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,4);
scatter(dual_del.Long_UTM,dual_del.Lat_UTM,4,dual_del.PRP21);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('Select PRP21 to be Deleted (press X to exit)');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
end
figure;
subplot(4,3,1);
scatter(dual_del.Long_UTM,dual_del.Lat_UTM,1,'r','.');
title('Raw Data');
rem = height(dual_dat_raw)-height(dual_del);
title(['Raw Data (removed: ' num2str(rem) ' points)']);

```

```

xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
format long;
assignin('base', 'dual_del', dual_del);
% -----%
% remove variables NaN & in-phase
dual_dat_r = dual_del;
dual_dat_r =
removevars(dual_dat_r,{'HCP05','PRP06','HCP05I','PRP06I','HCP10I','PRP11I','HCP20I','PRP21I'});
disp('remove in-phase')
size(dual_dat_r)
% -----%
% check for missing data
dual_dat_m = rmmissing(dual_dat_r);
disp('remove missing data');
size(dual_dat_m)
subplot(4,3,2);
rem = height(dual_dat_r)-height(dual_dat_m);
scatter(dual_dat_m.Long_UTM,dual_dat_m.Lat_UTM,1,'r','.');
title(['Missing Data (removed: ' num2str(rem) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_m', dual_dat_m);
% -----%
% Satellites < 4 delete
dual_dat_sat = dual_dat_m;
toDelete = dual_dat_sat.Satellites < 4;
dual_dat_sat(toDelete,:) = [];
disp('remove < 4 Satellites');
size(dual_dat_sat)
subplot(4,3,3);
rem = height(dual_dat_m)-height(dual_dat_sat);
scatter(dual_dat_sat.Long_UTM,dual_dat_sat.Lat_UTM,1,'r','.');
title(['Data with < 4 Satellites (removed: ' num2str(rem) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_sat', dual_dat_sat);
% -----%
% HDOP > 2 delete row
dual_dat_hdop = dual_dat_sat;
toDelete = dual_dat_hdop.HDOP > 2;
dual_dat_hdop(toDelete,:) = [];
disp('remove HDOP > 2');
size(dual_dat_hdop)
subplot(4,3,4);
rem = height(dual_dat_sat)-height(dual_dat_hdop);
scatter(dual_dat_hdop.Long_UTM,dual_dat_hdop.Lat_UTM,1,'r','.');
title(['Data with HDOP > 2 (removed: ' num2str(rem) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_hdop', dual_dat_hdop);
% -----%
% GPS quality < 2 delete
dual_dat_gps = dual_dat_hdop;
toDelete = dual_dat_gps.GPSQuality < 2;
dual_dat_gps(toDelete,:) = [];
disp('remove GPSQuality < 2');
size(dual_dat_gps)
subplot(4,3,5);
rem = height(dual_dat_hdop)-height(dual_dat_gps);
scatter(dual_dat_gps.Long_UTM,dual_dat_gps.Lat_UTM,1,'r','.');
title(['Data with GPS Quality < 2 (removed: ' num2str(rem) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_gps', dual_dat_gps);
% -----%
% remove row that contain negative mS/m
dual_dat_neg = dual_dat_gps;
toDelete = dual_dat_neg.HCP10 < 0;
dual_dat_neg(toDelete,:) = [];
toDelete = dual_dat_neg.PRP11 < 0;

```

```

dual_dat_neg(toDelete,:) = [];
toDelete = dual_dat_neg.HCP20 < 0;
dual_dat_neg(toDelete,:) = [];
toDelete = dual_dat_neg.PRP21 < 0;
dual_dat_neg(toDelete,:) = [];
disp('remove negative data');
size(dual_dat_neg)
subplot(4,3,6);
rem = height(dual_dat_gps)-height(dual_dat_neg);
scatter(dual_dat_neg.Long_UTM,dual_dat_neg.Lat_UTM,1,'r','.');
title(['Data with Negative Quadrature (removed: ' num2str(rem) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_neg', dual_dat_neg);
% -----%
% remove data > LIN assumption
dual_dat_lin = dual_dat_neg;
toDelete = dual_dat_lin.HCP10 > 720.3;
dual_dat_lin(toDelete,:) = [];
toDelete = dual_dat_lin.PRP11 > 5807.8;
dual_dat_lin(toDelete,:) = [];
toDelete = dual_dat_lin.HCP20 > 179.4;
dual_dat_lin(toDelete,:) = [];
toDelete = dual_dat_lin.PRP21 > 1593.7;
dual_dat_lin(toDelete,:) = [];
disp('remove data outside LIN');
size(dual_dat_lin)
subplot(4,3,7);
rem = height(dual_dat_neg)-height(dual_dat_lin);
scatter(dual_dat_lin.Long_UTM,dual_dat_lin.Lat_UTM,1,'r','.');
title(['Data Outside LIN (removed: ' num2str(rem) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_lin', dual_dat_lin);
% -----%
% check Dualem internal voltage : 8.5 V and 17 V good
dual_dat_v = dual_dat_lin;
toDelete = dual_dat_v.Voltage < 8.5;
dual_dat_v(toDelete,:) = [];
toDelete = dual_dat_v.Voltage > 17;
dual_dat_v(toDelete,:) = [];
disp('remove 8.5 < Voltage < 17');
size(dual_dat_v)
subplot(4,3,8);
rem = height(dual_dat_lin)-height(dual_dat_v);
scatter(dual_dat_v.Long_UTM,dual_dat_v.Lat_UTM,1,'r','.');
title(['Data with Bad Voltage (removed: ' num2str(rem) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_v', dual_dat_v);
% -----%
% temperature change > 2 oC delete
dual_dat_temp = dual_dat_v;
TempMed = median(dual_dat_temp.Temperature);
toDelete = abs(dual_dat_temp.Temperature - TempMed) > 2;
dual_dat_temp(toDelete,:) = [];
disp('remove Temp change > 2oC');
size(dual_dat_temp)
subplot(4,3,9);
rem = height(dual_dat_v)-height(dual_dat_temp);
scatter(dual_dat_temp.Long_UTM,dual_dat_temp.Lat_UTM,1,'r','.');
title(['Temperature change > 2\circC (removed: ' num2str(rem) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_temp', dual_dat_temp);
% -----%
% check roll, if > +-10 degree delete
dual_dat_roll = dual_dat_temp;
toDelete = abs(dual_dat_roll.Roll) > 10;
dual_dat_roll(toDelete,:) = [];
disp('remove roll > +-10 degree');

```

```

size(dual_dat_roll)
subplot(4,3,10);
rem = height(dual_dat_temp)-height(dual_dat_roll);
scatter(dual_dat_roll.Long_UTM,dual_dat_roll.Lat_UTM,1,'r','.');
title(['Data with Roll > \pm10\circ (removed: ' num2str(rem) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_roll', dual_dat_roll);
% -----
% remove duplicate data (Gator stop)
dual_dat_stop = dual_dat_roll;
toDelete=zeros(size(dual_dat_roll,1),1);
for n = 1:(size(dual_dat_roll,1)-1)
    if isequal(dual_dat_roll(n,1),dual_dat_roll(n+1,1)) &&...
        isequal(dual_dat_roll(n,2),dual_dat_roll(n+1,2))
        toDelete(n,1)=1;
    else
        toDelete(n,1)=0;
    end
end
toDelete(size(dual_dat_roll,1)-3:size(dual_dat_roll,1),1) = 1;
toDelete = logical(toDelete);
dual_dat_stop(toDelete,:) = [];
disp('remove Gator stop');
size(dual_dat_stop)
subplot(4,3,11);
rem = height(dual_dat_roll)-height(dual_dat_stop);
scatter(dual_dat_stop.Long_UTM,dual_dat_stop.Lat_UTM,1,'r','.');
title(['Duplicate Data Removed (removed: ' num2str(rem) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_stop', dual_dat_stop);
% -----
% remove spike data that exceed +- range (e.g. 5 mS/m)
dual_dat_sr = dual_dat_roll;
dual_sr_names = dual_dat_sr.Properties.VariableNames;
toDelete=zeros(size(dual_dat_sr,1),1);
m_dual_dat_sr = table2array(dual_dat_sr);
range = 5;
for j = 2 : (size(m_dual_dat_sr,1)-1)
    if (m_dual_dat_sr(j-1,5+5)+range < m_dual_dat_sr(j,5+5) && m_dual_dat_sr(j,5+5) >
m_dual_dat_sr(j+1,5+5)+range) &&...
        (m_dual_dat_sr(j-1,6+5)+range < m_dual_dat_sr(j,6+5) && m_dual_dat_sr(j,6+5) >
m_dual_dat_sr(j+1,6+5)+range) &&...
        (m_dual_dat_sr(j-1,7+5)+range < m_dual_dat_sr(j,7+5) && m_dual_dat_sr(j,7+5) >
m_dual_dat_sr(j+1,7+5)+range) &&...
        (m_dual_dat_sr(j-1,8+5)+range < m_dual_dat_sr(j,8+5) && m_dual_dat_sr(j,8+5) >
m_dual_dat_sr(j+1,8+5)+range)
        toDelete(j,1)=1;
    else
        toDelete(j,1)=0;
    end
end
toDelete = logical(toDelete);
m_dual_dat_sr(toDelete,:) = [];
disp('Remove spike +-5 mS/m');
size(m_dual_dat_sr)
dual_dat_s = array2table(m_dual_dat_sr);
dual_dat_s.Properties.VariableNames = dual_sr_names;
subplot(4,3,12);
rem = height(dual_dat_roll)-height(dual_dat_s);
scatter(dual_dat_s.Long_UTM,dual_dat_s.Lat_UTM,1,'r','.');
title(['Remove spike \pm' num2str(range) ' mS/m (removed: ' num2str(rem) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_s', dual_dat_s);
% -----
% moving median smoothing for all coil
% display before smoothing
figure;
subplot(2,2,1);

```

```

scatter(dual_dat_s.Long_UTM,dual_dat_s.Lat_UTM,4,dual_dat_s.HCP10);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('HCP10 before median smoothing');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,2);
scatter(dual_dat_s.Long_UTM,dual_dat_s.Lat_UTM,4,dual_dat_s.PRP11);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('PRP11 before median smoothing');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,3);
scatter(dual_dat_s.Long_UTM,dual_dat_s.Lat_UTM,4,dual_dat_s.HCP20);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('HCP20 before median smoothing');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,4);
scatter(dual_dat_s.Long_UTM,dual_dat_s.Lat_UTM,4,dual_dat_s.PRP21);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('PRP21 before median smoothing');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
dual_dat_mr = dual_dat_s;
% convert HCP1,PRP11,HCP2,PRP21 to matrix to use movmedian and maintain coordinate
m_dual_mr = table2array(dual_dat_mr(:,10:13));
med_window = 3;
med = movmedian(m_dual_mr,med_window,1);
med = array2table(med);
dual_dat_mr(:,10:13)=med;
dual_dat_med = dual_dat_mr;
disp('Median Filter (3 windows, +-5 mS/m)');
size(dual_dat_med)
% display after smoothing
figure;
subplot(2,2,1);
scatter(dual_dat_med.Long_UTM,dual_dat_med.Lat_UTM,4,dual_dat_med.HCP10);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('HCP10 after median smoothing');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,2);
scatter(dual_dat_med.Long_UTM,dual_dat_med.Lat_UTM,4,dual_dat_med.PRP11);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('PRP11 after median smoothing');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,3);
scatter(dual_dat_med.Long_UTM,dual_dat_med.Lat_UTM,4,dual_dat_med.HCP20);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('HCP20 after median smoothing');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,4);
scatter(dual_dat_med.Long_UTM,dual_dat_med.Lat_UTM,4,dual_dat_med.PRP21);
colbar = colorbar;
colbar.Label.String = 'mS/m';

```

```

c.Location = 'eastoutside';
title('PRP21 after median smoothing');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_med', dual_dat_med);
% -----
% Remove headland turns
dual_med_names = dual_dat_med.Properties.VariableNames;
strt = 1;
T = NaN(size(dual_dat_med,1),size(dual_dat_med,2));
Tsect = table2array(dual_dat_med);
T(strt,:) = Tsect(strt,:);
flag = true;
mfit = zeros(size(Tsect,1),1);
mfit2 = zeros(size(Tsect,1),1);
strangle = zeros(size(Tsect,1),3);
D_headland = NaN(size(Tsect,1),size(Tsect,2));
for i = strt+1 : (size(Tsect,1)-3)
    if flag
        T(i,:) = Tsect(i,:);
        x = Tsect(strt:i+3,1);
        y = Tsect(strt:i+3,2);
        fit = polyfit(x,y,1);
        mfit(i,1) = fit(1);
    end
    x2 = Tsect(i:i+1,1);
    y2 = Tsect(i:i+1,2);
    fit2 = polyfit(x2,y2,1);
    mfit2(i,1) = fit2(1);
    angle1 = atand(fit(1));
    angle2 = atand(fit2(1));
    angle = abs(angle1-angle2);
    strangle(i,1)=angle1;
    strangle(i,2)=angle2;
    strangle(i,3)=angle;
    if angle > 15
        flag = false;
        strt = i;
        D_headland(i,:) = Tsect(i+1,:);
    else
        flag = true;
    end
end
T(1:2,:)=NaN;
D_headland(1:2,:)=[];
% delete last 5 data : end of field
T(size(T,1)-5:size(T,1),:)=NaN;
T = rmmissing(T);
dual_dat_tsect = array2table(T);
dual_dat_tsect.Properties.VariableNames = dual_med_names;
figure;
rem = height(dual_dat_med)-height(dual_dat_tsect);
scatter(dual_dat_tsect.Long_UTM,dual_dat_tsect.Lat_UTM,4,'r','o');
title(['Remove Headland Turning (removed: ' num2str(rem) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_tsect', dual_dat_tsect);
% -----
% decimation
% calculate transect spacing
posi = [true, isnan(D_headland(:, 1)).', true];
init = strfind(posi, [true, false]);
fina = strfind(posi, [false, true]) - 1;
C_decim = cell(1, length(init));
for i_dec = 1:length(init)
    C_decim{i_dec} = D_headland(init(i_dec):fina(i_dec), :);
end
% calculate distance first 5 rows and average them
U_dist_1 = [C_decim{1}];
Row1 = vertcat(U_dist_1(1,1:2),U_dist_1(size(U_dist_1,1),1:2));
Eucl1 = pdist(Row1, 'euclidean'); %this when operator just started (plenty steering adjustment)

```

```

                                %so, does not count as turning
U_dist_2 = [C_decim{2}];
Row2 = vertcat(U_dist_2(1,1:2),U_dist_2(size(U_dist_2,1),1:2));
Eucl2 = pdist(Row2, 'euclidean'); %this when the first turning started
U_dist_3 = [C_decim{3}];
Row3 = vertcat(U_dist_3(1,1:2),U_dist_3(size(U_dist_3,1),1:2));
Eucl3 = pdist(Row3, 'euclidean');
U_dist_4 = [C_decim{4}];
Row4 = vertcat(U_dist_4(1,1:2),U_dist_4(size(U_dist_4,1),1:2));
Eucl4 = pdist(Row4, 'euclidean');
U_dist_5 = [C_decim{5}];
Row5 = vertcat(U_dist_5(1,1:2),U_dist_5(size(U_dist_5,1),1:2));
Eucl5 = pdist(Row5, 'euclidean');
Row = (Eucl2+Eucl3+Eucl4+Eucl5)/4;
% decimation started
dual_tsect_names = dual_dat_tsect.Properties.VariableNames;
m_dual_dat_tsect = table2array(dual_dat_tsect);
m_dual_dec = m_dual_dat_tsect;
m = 1; %initialize start from first data
for k = 2: (size(m_dual_dec ,1))
    dist = vertcat(m_dual_dec(m,1:2),m_dual_dec(k,1:2));
    Eucl_dist = pdist(dist, 'euclidean');
    if Eucl_dist < Row
        m_dual_dec(k,:) = NaN;
    end
    if Eucl_dist >= Row
        m = k;
    end
end
m_dual_dec = rmmissing(m_dual_dec);
rem_dec = size(dual_dat_tsect,1)-size(m_dual_dec,1);
disp(['Decimation Deletion : ' num2str(rem_dec) ' points']);
dual_dat_dec = array2table(m_dual_dec);
dual_dat_dec.Properties.VariableNames = dual_tsect_names;
figure;
scatter(dual_dat_dec.Long_UTM,dual_dat_dec.Lat_UTM,4,'r','o');
rem = height(dual_dat_tsect)-height(dual_dat_dec);
title(['Decimated Data (' num2str(round(Row,3,'significant')) ' m interrow, removed:'
num2str(rem) ' points')]);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
assignin('base', 'dual_dat_dec', dual_dat_dec);
% -----%
% check and delete additional data manually ;press x to exit
dual_del_end = dual_dat_dec;
figure;
scatter(dual_del_end.Long_UTM,dual_del_end.Lat_UTM,4);
title('Select Points to be Deleted (press X to exit)');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
while true
    w = waitforbuttonpress;
    switch w
        case 1 %(keyboard press)
            key = get(gcf,'currentcharacter');
            switch key
                case 120 % 120 ASCII for lowercase x
                    close();
                    break;
            end
        end
        pl = selectdata('selectionmode','closest','action','delete');
        dual_del_end(pl,:) = [];
    scatter(dual_del_end.Long_UTM,dual_del_end.Lat_UTM,4);
    title('Select Points to be Deleted (press X to exit)');
    xlabel('Normalized Easting (m)');
    ylabel('Normalized Northing (m)');
end
figure;
rem_end = height(dual_dat_raw)-height(dual_del_end);
scatter(dual_del_end.Long_UTM,dual_del_end.Lat_UTM,4,'r','o');

```

```

title(['Cleaned Data (total: ' num2str(height(dual_del_end)) 'points // total removed: '
num2str(rem_end) ' points)']);
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
format long;
assignin('base', 'dual_del_end', dual_del_end);
% -----%
% draw final
figure;
subplot(2,2,1);
scatter(dual_del_end.Long_UTM,dual_del_end.Lat_UTM,4,dual_del_end.HCP10);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('HCP10 Filtered');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,2);
scatter(dual_del_end.Long_UTM,dual_del_end.Lat_UTM,4,dual_del_end.PRP11);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('PRP11 Filtered');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,3);
scatter(dual_del_end.Long_UTM,dual_del_end.Lat_UTM,4,dual_del_end.HCP20);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('HCP20 Filtered');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
subplot(2,2,4);
scatter(dual_del_end.Long_UTM,dual_del_end.Lat_UTM,4,dual_del_end.PRP21);
colbar = colorbar;
colbar.Label.String = 'mS/m';
c.Location = 'eastoutside';
title('PRP21 Filtered');
xlabel('Normalized Easting (m)');
ylabel('Normalized Northing (m)');
% -----%
% remove cleaned unused variables :
% Satellites, HDOP, GPSQuality, Voltage, Temperature, Pitch, Roll Longitude and Latitude EXT
dual_dat_cln = dual_del_end;
dual_dat_cln =
removevars(dual_dat_cln,{'Long_UTM','Lat_UTM','Elevation','Satellites','HDOP','GPSQuality','Volta
ge','Temperature','Pitch','Roll','LongitudeExt','LatitudeExt'});
disp('remove cleaned variables');
size(dual_dat_cln)
assignin('base', 'dual_dat_cln', dual_dat_cln);
dual_dat_noDec = dual_dat_tsect;
dual_dat_noDec =
removevars(dual_dat_noDec,{'Long_UTM','Lat_UTM','Elevation','Satellites','HDOP','GPSQuality','Vol
tage','Temperature','Pitch','Roll','LongitudeExt','LatitudeExt'});
disp('remove cleaned variables');
size(dual_dat_noDec)
assignin('base', 'dual_dat_noDec', dual_dat_noDec);
% -----%
% save
dual_dat_cln.Properties.VariableNames = {'Longitude', 'Latitude', 'Time_stamp', 'HCP1', 'PRP1',
'HCP2', 'PRP2'};
writetable(dual_dat_cln,'Duaem_Filtered.csv','Delimiter', ',', 'QuoteStrings',true);
dual_dat_show = num2str(table2array(dual_dat_cln));
set(handles.result_text,'String',dual_dat_show); %show the data
dual_dat_noDec.Properties.VariableNames = {'Longitude', 'Latitude', 'Time_stamp', 'HCP1', 'PRP1',
'HCP2', 'PRP2'};
writetable(dual_dat_noDec,'Duaem_Filtered_noDecimate.csv','Delimiter', ',', 'QuoteStrings',true);

```

9.7. Brute-Force EC_a Inversion Code

```
function varargout = GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm(varargin)
% GUI_5_BRUTE_FORCE_10_OUT1TABLE_BARU_OPT2RES_TO_200MSM MATLAB code for
GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm.fig
% GUI_5_BRUTE_FORCE_10_OUT1TABLE_BARU_OPT2RES_TO_200MSM, by itself, creates a new
GUI_5_BRUTE_FORCE_10_OUT1TABLE_BARU_OPT2RES_TO_200MSM or raises the existing
% singleton*.
%
% H = GUI_5_BRUTE_FORCE_10_OUT1TABLE_BARU_OPT2RES_TO_200MSM returns the handle to a new
GUI_5_BRUTE_FORCE_10_OUT1TABLE_BARU_OPT2RES_TO_200MSM or the handle to
% the existing singleton*.
%
%
GUI_5_BRUTE_FORCE_10_OUT1TABLE_BARU_OPT2RES_TO_200MSM('CALLBACK',hObject,eventData,handles,...)
calls the local
% function named CALLBACK in GUI_5_BRUTE_FORCE_10_OUT1TABLE_BARU_OPT2RES_TO_200MSM.M with
the given input arguments.
%
% GUI_5_BRUTE_FORCE_10_OUT1TABLE_BARU_OPT2RES_TO_200MSM('Property','Value',...) creates a
new GUI_5_BRUTE_FORCE_10_OUT1TABLE_BARU_OPT2RES_TO_200MSM or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm_OpeningFcn
gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to
GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm

% Last Modified by GUIDE v2.5 31-Jul-2017 12:49:01

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',      mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm_OpeningFcn, ...
                  'gui_OutputFcn',  @GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm is made visible.
function GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm_OpeningFcn(hObject,~, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm (see
VARARGIN)
```

```

% Choose default command line output for GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUI_5_brute_force_10_Out1Table_baru_Opt2res_to_200mSm_OutputFcn(~, ~,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in dual_path_but.
function dual_path_but_Callback(~, ~, handles)
% hObject      handle to dual_path_but (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
[dual_filename, dual_pathname] = uigetfile (*.csv','Dualem data');
dual_fullpathname = strcat(dual_pathname,dual_filename);
dual_file = fileread(dual_fullpathname);
set(handles.dual_path_text2,'String',dual_fullpathname); %show fullpathname
set(handles.dual_path_text3,'String',dual_file); %show the data
% dual_dat_filename = dual_fullpathname;
%guidata(hObject, handles);
assignin('base', 'dual_dat_filename', dual_filename);
assignin('base', 'pathname', dual_pathname);

% --- Executes on button press in cum_res_shallow.
function cum_res_shallow_Callback(~, ~, handles)
% hObject      handle to cum_res_shallow (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
[cum_shallow_filename, cum_shallow_pathname] = uigetfile (*.csv','Cumulative response data
shallow');
cum_shallow_fullpathname = strcat(cum_shallow_pathname,cum_shallow_filename);
cum_shallow_file = fileread(cum_shallow_fullpathname);
set(handles.cum_shallow_path_text2,'String',cum_shallow_fullpathname); %show fullpathname
set(handles.cum_shallow_path_text3,'String',cum_shallow_file); %show the data
% cum_dat_filename = cum_fullpathname;
%guidata(hObject, handles);
assignin('base', 'cum_shallow_dat_filename', cum_shallow_filename); % https://goo.gl/yEccSz
https://goo.gl/pw33Vo

% --- Executes on button press in cum_res_deep.
function cum_res_deep_Callback(~, ~, handles)
% hObject      handle to cum_res_deep (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
[cum_deep_filename, cum_deep_pathname] = uigetfile (*.csv','Cumulative response data deep');
cum_deep_fullpathname = strcat(cum_deep_pathname,cum_deep_filename);
cum_deep_file = fileread(cum_deep_fullpathname);
set(handles.cum_deep_path_text2,'String',cum_deep_fullpathname); %show fullpathname
set(handles.cum_deep_path_text3,'String',cum_deep_file); %show the data
% cum_dat_filename = cum_fullpathname;
%guidata(hObject, handles);

```

```

assignin('base', 'cum_deep_dat_filename', cum_deep_filename); % https://goo.gl/yEccSz
https://goo.gl/pw33Vo

function start_close_Callback(~, ~, handles)
% hObject      handle to start_close (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
from = str2double (get(handles.from,'string'));
assignin('base', 'from', from);
to = str2double (get(handles.to,'string'));
assignin('base', 'to', to);
%inversion with ALL coil HCP1,2 PRP1,2 and depth not counting the sled
%CTRL+C --> stop run
%CTRL+L --> clear command window
%F9 evaluate (run) section
%clear workspace -----%
tic; %https://goo.gl/3Zzu45
% waitbar https://goo.gl/izvqWc https://goo.gl/rSGYRB
h = waitbar(0,'1','Name','Progress...',...
    'CreateCancelBtn',...
    'setappdata(gcf,'canceling',1)');
setappdata(h,'canceling',0)
%-----%
% read data https://goo.gl/Pc3Z8I -----
% dualem
pathname = evalin('base','pathname');
dual_dat_filename = evalin('base', 'dual_dat_filename');
cum_shallow_dat_filename = evalin('base', 'cum_shallow_dat_filename');
cum_deep_dat_filename = evalin('base', 'cum_deep_dat_filename');
cd(pathname);
dual_dat = readtable(dual_dat_filename, 'Delimiter',' ','ReadVariableNames',true);
%re-arrange dual data to : Long, Lat, Time_stamp, PRP1, PRP2, HCP1, HCP2
dual_dat_arr = [dual_dat(:,1:3) dual_dat(:,5) dual_dat(:,7) dual_dat(:,4) dual_dat(:,6)];
%convert to matrix
dual_dat_m = table2array(dual_dat_arr); %make dualem matrix https://goo.gl/7Ozz4A
% cumulative response
% shallow
cum_dat_s = readtable(cum_shallow_dat_filename, 'Delimiter',' ','ReadVariableNames',true);
%convert to matrix
cum_dat_m_s = table2array(cum_dat_s); %make cum matrix https://goo.gl/7Ozz4A
% deep
cum_dat_d = readtable(cum_deep_dat_filename, 'Delimiter',' ','ReadVariableNames',true);
%convert to matrix
cum_dat_m_d = table2array(cum_dat_d); %make cum matrix https://goo.gl/7Ozz4A
% create final combination dualem data + brute force inversion; 12 kolom
(Long,Lat,Time_stamp,PRP1,PRP2,HCP1,HCP2,depth,n,sigma_shallow,sigma_deep,RMSE)
V = zeros(height(dual_dat),12);
% first data point to calculate
first_point = evalin('base','from');
% last data point to calculate
last_point = evalin('base','to');
% range calculation
range_points = last_point-first_point;
% scale bar e.g.100%
mini = 1;
maxi = 100;
range_minmax = maxi-mini;
% extract dual data https://goo.gl/PVxkzD
for i = first_point:last_point
    scale_pos = round((maxi-((range_minmax*(last_point-i))/range_points)),0);
%https://goo.gl/MJdr4n
    waitbar(scale_pos/range_minmax,h,sprintf('%d%%',scale_pos))
    if getappdata(h,'canceling')
        break
    end
%    dual_table = dual_dat_arr(i, 4:7); %dualem data table for writing to excel
    dual = dual_dat_m(i,4:7);
    dualr = dual.'; %transpose dualr = real data
%    dualr_range = max(dualr);
    % extract cum data
    m = size(cum_dat_s,1); %number of depth increment (10-150)

```

```

R = zeros(m,5); %min RMSE mat; number of row = 15 depth increment, number of column = 5
(depth,n,teta_shallow,teta_deep,RMSE)
for j = 1 : m
    cumd_s = cum_dat_m_s(j, 2:5); % select cumulative response PRP1,2,HCP1,2
    cum_dual1 = cumd_s.';
    cumd_d = cum_dat_m_d(j, 2:5);
    cum_dual2 = cumd_d.';
    cum_dual = horzcat(cum_dual1,cum_dual2);
    q = 2000; %set jumlah step ; 0.2 sampe 200 tiap step naik 0.2 total ada 1000 step
    untuk tiap teta (ada 2 teta total 1 juta)
        n = q/2*q/2; %set total iteration brute force
        C = zeros(n,4); %number of row = n (iteration), number of column = 4 (iteration, teta
        shallow, teta deep, RMSE)
        % start brute force
        for t1 = 2 :2: q
            teta1 = t1/10;
            for t2 = 2 :2: q %resolution 0.2 mS/m ; step from 0.2 to 200 mS/m
                teta2 = t2/10;
                teta = vertcat(teta1,teta2); %join matrix
                % RMSE
                dualp = cum_dual*teta;
                err = dualp - dualr;
                SE = err.*err; %https://goo.gl/A6x674
                sum_SE = sum(SE); %https://goo.gl/mQaB0f
                MSE = sum_SE/size(SE,1); %https://goo.gl/bpvklm
                RMSE = sqrt(MSE);
                it = (t1-(t1/2+1))*q/2+(t2/2); %iteration per 0.2
                C(it,1)=it;
                C(it,2)=teta(1,1);
                C(it,3)=teta(2,1);
                C(it,4)=RMSE;
            end
        end
        T = array2table(C); %https://goo.gl/4FAC5q
        T.Properties.VariableNames = {'n', 'teta_shallow', 'teta_deep', 'RMSE'};
        %https://goo.gl/00rpCR
        [~,I]=min(C); %https://goo.gl/G03Gsz --> search for smallest elements in each column
        and its row indices; matrix M contains the smallest element in each column; matrix I contains the
        row number of the smallest element in each column
        min_RMSE = C(I(1,4),:); %I(1,4)--> select baris ke-1 kolom ke-4 (min RMSE) di matrix
        C karena I(index data RMSE di kolom ke-4)
        min_RMSE_mat = array2table(min_RMSE); %https://goo.gl/4FAC5q
        depth = table2array(cum_dat_s(j,1)); %https://goo.gl/jJZlIT https://goo.gl/mRZ085
        R(j,1)=depth;
        R(j,2)=min_RMSE(1,1); %number of iteration
        R(j,3)=min_RMSE(1,2); %teta1 (shallow)
        R(j,4)=min_RMSE(1,3); %teta2 (deep)
        R(j,5)=min_RMSE(1,4); %min RMSE
    end
    %select the smallest RMSE with its depth and combine in one table
    [~,J]=min(R);
    comb_min_RMSE = R(J(1,5),:); %J(1,5)--> select baris ke-1 kolom ke-5 (min RMSE dari 1
    sampling point) di matrix R karena I(index data RMSE di kolom ke-5)
    V(i,1:7) = dual_dat_m(i,1:7);
    V(i,8:12) = comb_min_RMSE;
    V_out = num2str(V(i,1:12),3); %https://goo.gl/pKnE7z
    set(handles.result,'string',V_out);
end
delete(h);
F = array2table(V);
F.Properties.VariableNames = {'Long', 'Lat', 'Time_stamp', 'PRP1', 'PRP2', 'HCP1', 'HCP2',
'depth', 'n', 'teta_shallow', 'teta_deep', 'RMSE'};
writetable(F,'GUI 5.xlsx','Range','A1');
% writetable(T,'abcrandom.xlsx','Sheet',i,'Range','A1');
msgbox('Operation Completed'); %https://goo.gl/Y9wkpu
toc;

function from_Callback(~, ~, ~)
% hObject handle to from (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of from as text
%         str2double(get(hObject,'String')) returns contents of from as a double

% --- Executes during object creation, after setting all properties.
function from_CreateFcn(hObject, ~, ~)
% hObject    handle to from (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function to_Callback(~, ~, ~)
% hObject    handle to to (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of to as text
%         str2double(get(hObject,'String')) returns contents of to as a double

% --- Executes during object creation, after setting all properties.
function to_CreateFcn(hObject, ~, ~)
% hObject    handle to to (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```