

Technical Report

TR-20-01

December 2020



Probabilistic model for the pitting of copper canisters under aerobic, saturated conditions

Scott Briggs
Christina Lilja
Fraser King

SVENSK KÄRNBRÄNSLEHANTERING AB

SWEDISH NUCLEAR FUEL
AND WASTE MANAGEMENT CO

Box 3091, SE-169 03 Solna
Phone +46 8 459 84 00
skb.se

SVENSK KÄRNBRÄNSLEHANTERING

ISSN 1404-0344

SKB TR-20-01

ID 1883736

December 2020

Probabilistic model for the pitting of copper canisters under aerobic, saturated conditions

Scott Briggs, Nuclear Waste Management Organization

Christina Lilja, Svensk Kärnbränslehantering AB

Fraser King, Integrity Corrosion Consulting Ltd.

This report is published on www.skb.se

© 2020 Svensk Kärnbränslehantering AB

Figure no	Credits
2-2	Reprinted from Journal of The Electrochemical Society, Cong H , Michels H T, Scully J R, Passivity and Pit Stability Behavior of Copper as a Function of Selected Water Chemistry Variables (2009). © The Electrochemical Society. Reproduced by permission of IOP Publishing Ltd. All rights reserved.
2-4	Reprinted from MRS Online Proceedings Library, Imai H, Fukuda T, Akashi M, Effects of anionic species on the polarization behaviour of copper for waste package material in artificial ground water, (1996), with permission from Elsevier.

Summary

The development of a probabilistic model to predict the extent of pitting corrosion of copper canisters is described. The model accounts for not only the stochastic nature of pitting corrosion but also the variability and uncertainty in the repository environment and in how it evolves with time. Because of the availability of mechanistic information and of suitable input data, the model was developed on the assumption of aerobic, saturated conditions in the near field.

The model is based on well-established criteria for pit initiation and propagation. Pit initiation requires that the canister surface is passive and that the value of the corrosion potential exceeds that of the film breakdown potential. Propagation requires that the corrosion potential exceeds the value of the repassivation potential. Values for these two critical potentials, as well as information on the conditions under which passivation can be expected, are available in the literature. The environmental parameters of interest are the temperature, bentonite pore-water pH, and the pore-water concentrations of chloride, sulphate, and bicarbonate ions, which are described by suitable distributions.

The probabilistic model was developed using the commercially available MATLAB software. Various classification and regression algorithms were used to determine the active-passive nature of the surface and to fit the critical potential data, respectively. A single run, or realization, represents the evolution of the pitting behaviour of the canister in an individual deposition hole over the period of interest, typically the first 100 years after emplacement. A specific pH, chloride concentration, and temperature-time profile was sampled for each realization. The concentrations of sulphate and bicarbonate ions are solubility-limited and vary with time due to the variation of temperature. At each time step during the execution of the code, the classification algorithm is used to determine the nature of the canister surface by comparing the modelled environmental conditions to those found experimentally to lead to either active or passive behaviour. For passive conditions, the possibility of pit initiation and, in the event of initiation, the duration of propagation are determined using the regressed critical potential data. The pit depth is estimated using the predicted duration of pitting and an empirical pit growth expression.

The full probabilistic analysis comprises a total of one million realizations, representing an average of 165 realizations for each of the approximately 6000 canisters in the repository. The maximum pit depth is predicted to be approximately 1 mm. The overwhelming majority of pits are less than 100 μm in depth, which corresponds to the corrosion allowance used for localised corrosion of the canisters for the SR-Site safety assessment.

Although it is recognized that, depending on the relative rates of saturation and oxygen depletion, aerobic, saturated conditions may not exist in the repository, the probabilistic approach developed will be useful for assessing localised corrosion under unsaturated, aerobic conditions in the future.

Contents

1	Introduction	7
2	Background	9
2.1	Passivation	9
2.2	Pit initiation and growth	12
2.3	Criteria for pit initiation and propagation	13
3	Input data	15
3.1	Corrosion and electrochemical data	15
3.1.1	Conditions for active or passive behaviour	15
3.1.2	Critical potentials	16
3.1.3	Pit growth kinetics	17
3.1.4	Corrosion potential	18
3.1.5	Applicability of electrochemical and pit growth data	19
3.2	Environmental data	20
3.2.1	Canister temperature	20
3.2.2	Pore-water pH	21
3.2.3	Chloride concentration	23
3.2.4	Sulphate and bicarbonate concentrations	24
3.2.5	Oxygen concentration	26
4	Implementation of the probabilistic pitting model	29
4.1	Structure of the model	29
4.1.1	Control flow chart	29
4.1.2	Environment flow chart	31
4.1.3	Pit Initiation flow chart	32
4.1.4	Pit Propagation flow chart	33
4.2	Software and hardware	34
4.3	Model development	34
4.3.1	Classification methods for the active-passive data	34
4.3.2	Regression methods for critical potential data	39
4.3.3	Pit growth kinetics	43
4.3.4	Time discretisation	45
4.4	Final version of the model as implemented	46
4.4.1	Sources and treatment of input data	46
4.4.2	Simulations performed	47
5	Results of probabilistic pitting analysis	49
5.1	Full probabilistic analysis	49
5.2	Visualization of pitting behaviour based on 100 realizations	53
5.3	Progression of pitting behaviour	60
5.4	Effect of maximum bentonite pore-water pH	63
5.5	Comparison of replicate simulations	64
5.6	Scaling issues	64
6	Conclusions	67
7	Acknowledgements	69
	References	71
Appendix	Source Code	75

1 Introduction

Copper canisters in a planned KBS-3 repository at the Forsmark site in Sweden are expected to be subject to general (uniform) corrosion and a limited degree of localised attack (King and Lilja 2014, King et al. 2010, SKB 2010). Based on the results of both short-term laboratory experiments and longer-term full-scale *in situ* tests at the Äspö Hard Rock Laboratory, the localised attack is expected to take the form of surface roughening rather than discrete pitting corrosion (King et al. 2010). The nature of the near-surface environment is expected to result in an active copper surface rather than the passive state required for pit initiation and growth.

Although the canister is expected to be exposed to predominantly active conditions, it is nevertheless prudent to assess the probability of pitting corrosion. Pitting is an inherently stochastic process (Szklańska-Smiałowska 2005) which makes a probabilistic approach appropriate (Scully and Hicks 2012, Scully and Edwards 2013, Scully et al. 2016). In addition, there are other sources of uncertainty and variability to take into account, such as the variability in the canister surface environment from one deposition hole to another and uncertainty in the conditions that produce a passive surface.

It is well understood that the repository environment evolves with time and that this evolution impacts the nature and time dependence of different corrosion processes (King et al. 2017). Pitting of copper is usually associated with aerobic conditions (King et al. 2010), although it has recently been suggested that passive Cu_2S films susceptible to localised breakdown could form during the long-term anaerobic phase (Kong et al. 2017a, b, Macdonald et al. 2016, Mao et al. 2014). Non-uniform wetting of the canister surface during the evolution of the repository saturation may also facilitate the spatial separation of anodic and cathodic processes, leading to localised attack. Therefore, for the purposes of assessing the possible pitting of copper canisters, the evolution of the repository is characterised by four possible conditions:

- Aerobic, unsaturated conditions.
- Aerobic, saturated conditions.
- Anaerobic, unsaturated conditions.
- Anaerobic, saturated conditions.

The repository environment will start out aerobic and unsaturated and will eventually become saturated and anaerobic. The existence, or not, of either aerobic, saturated or of anaerobic, unsaturated conditions will depend on the relative rates of saturation and evolution of the redox conditions.

Here we describe the development of a probabilistic pitting model for aerobic, saturated conditions. The aims of the work are to develop the methodology for probabilistic pitting assessments and to apply that methodology to this particular set of environmental conditions. In doing so, we acknowledge that aerobic, saturated conditions may be unlikely to occur if saturation is slow or O_2 consumption is fast, but we have chosen these conditions because of the availability of input data for the model. Having developed the methodology, the intention is to assess the probability of pitting under the other sets of environmental conditions in future studies.

This report describes the development of the probabilistic methodology and its application to the assessment of pitting of copper canisters under aerobic, saturated conditions. A brief review of the literature on the passivation and pitting of copper in relevant environments is given in Section 2, primarily in order to define the nature of the required input data for the model. The required corrosion, electrochemical, and environmental input data are summarised in Section 3, including the nature of the distribution functions for the probabilistic assessment. During the development of the model, the use of various statistical classification and regression techniques were investigated using different sets of input data. The outcome of these trials are described in Section 4, along with a definition of the final versions of the model and input data used for the probabilistic analysis itself. Various results of the probabilistic assessment are presented in Section 5, including the outcome of the complete simulations comprising one million individual runs or realizations, as well as the results of limited simulations of 100 realizations which are useful for visualising trends in the analyses.

2 Background

2.1 Passivation

Pitting requires that the surface be in the passive condition (Frankel and Sridhar 2008, King and Lilja 2014). There is a general consensus that the passivation of copper is promoted by increasing pH and/or $[\text{HCO}_3^-]$, while increasing $[\text{Cl}^-]$ and $[\text{SO}_4^{2-}]$ support active dissolution (Cong et al. 2009, Imai et al. 1996, Qin et al. 2017). Increasing temperature promotes active dissolution (Qin et al. 2017).

While there is no formal definition of what constitutes a passive surface, the distinction between active (A) and passive (P) behaviour can be usefully distinguished using cyclic voltammetry (Cong et al. 2009, Imai et al. 1996, Qin et al. 2017). Figure 2-1 shows examples of A and P behaviour for copper determined by cyclic voltammetry (Qin et al. 2017). There are, of course, intermediate cases with characteristics of both A and P behaviour which represent the border between the two behaviours. Thus, the boundary between A and P behaviour is not necessarily well defined and represents another source of uncertainty in the analysis.

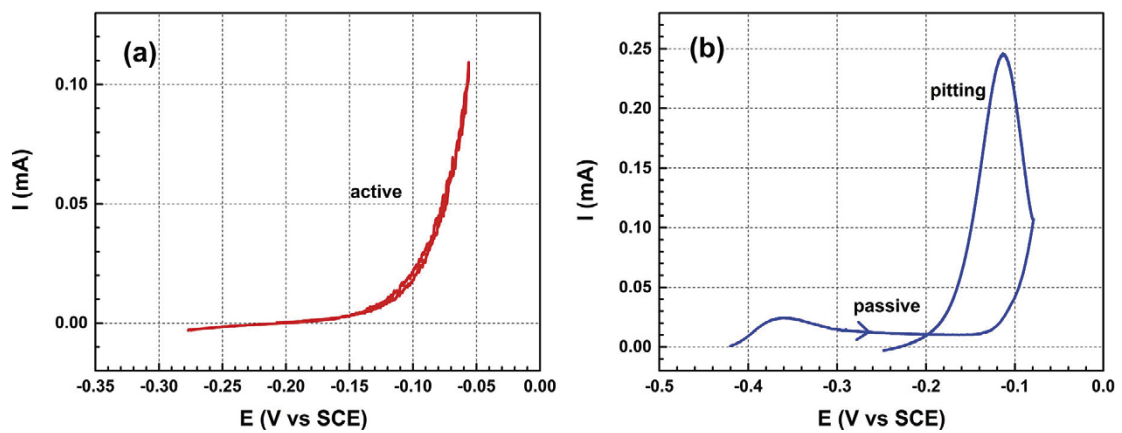


Figure 2-1. Examples of active (left) and passive (right) cyclic voltammograms for copper (Qin et al. 2017).

Passivation of copper can be the result of the formation of both a copper(I) and copper(II) surface film. Figure 2-2 shows the formation of both Cu(I) and Cu(II) species associated with the passivation of copper at pH 8.5 in borate buffer in the absence of any other passivating or activating anions (Cong et al. 2009). Upon scanning the potential from negative to more-positive values, the onset of passivation is associated with the formation of a Cu_2O film, but with increasing polarisation a duplex $\text{Cu}_2\text{O}/\text{Cu(II)}$ film (either CuO or Cu(OH)_2) is formed. Thus the degree of passivation also depends on potential which, under freely corroding conditions, is determined by the corrosion potential (E_{CORR}) which, in turn, is determined in part by the redox conditions.

Qin et al. (2017) conducted a comprehensive survey of the effects of temperature, pH, and of chloride, sulphate, and total carbonate concentrations on the A/P behaviour of copper. Figure 2-3 shows examples of the effect of different anions as a function of pH and of the overall A/P “maps” that were developed and which define different regions of activating or passivating environmental conditions. Imai et al. (1996) developed similar maps, an example of which is shown in Figure 2-4. Unlike Qin et al. (2017), Imai et al. (1996) did not systematically vary the pH, and did not report the pH of the solutions used, which presumably varied with $[\text{HCO}_3^-]$. Thus, the increasing tendency towards passivation with increasing $[\text{HCO}_3^-]$ shown in the figure is at least partly the effect of the increase in pH. Imai et al. (1996) also studied the effect of dissolved O_2 (DO) concentration on passivation, even though the potential was controlled and scanned during the cyclic voltammetric measurement. Passivation was found to be promoted by higher DO concentrations (Figure 2-4), which may have resulted from the homogenous oxidation of Cu(I) to Cu(II) by dissolved O_2 and the resulting enhancement of passivity.

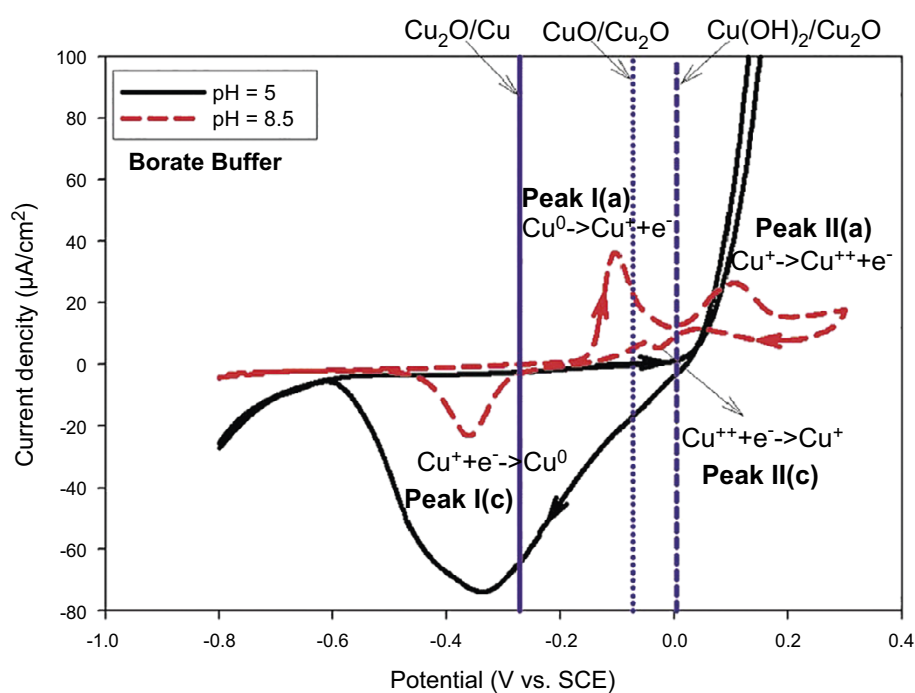


Figure 2-2. Cyclic voltammograms on copper in borate buffer at pH 5 (active) and pH 8.5 (passive) at room temperature, along with the reversible potentials for various copper solid equilibria at pH 8.5 (Cong et al. 2009).

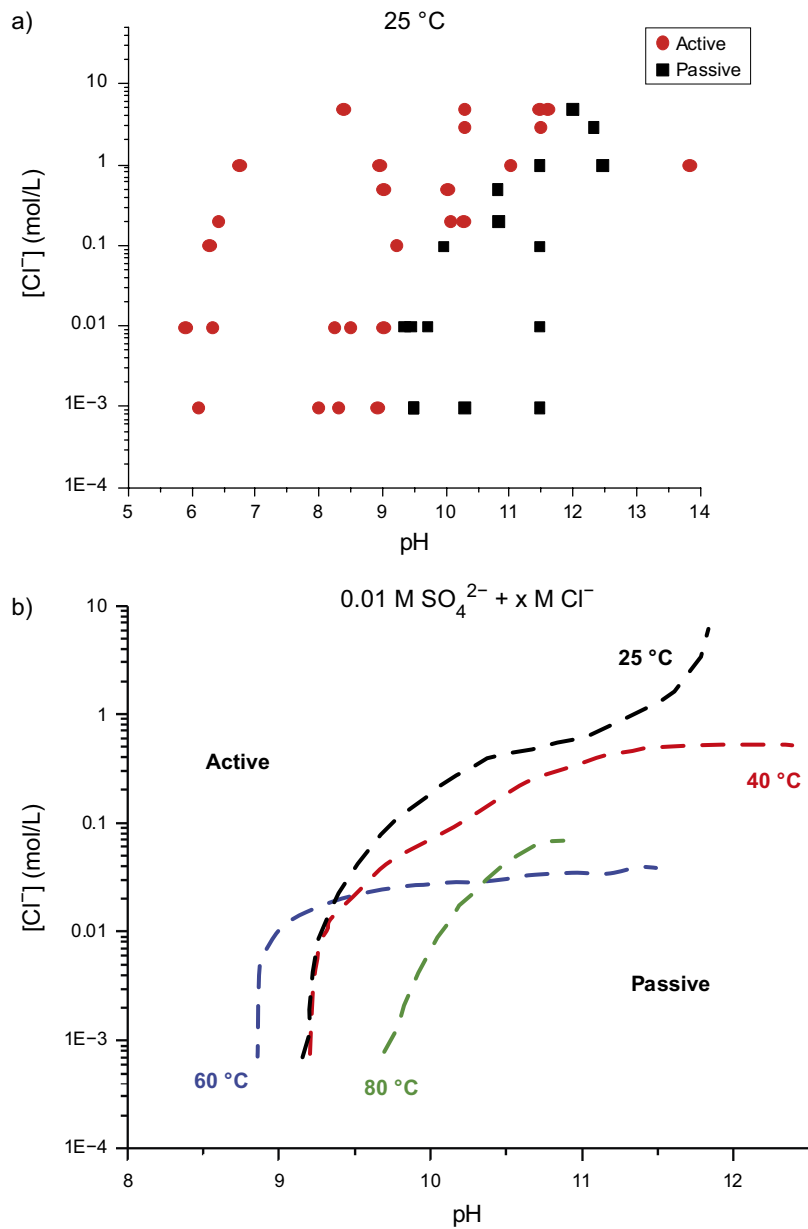


Figure 2-3. Examples of the effects of environmental conditions on the A/P behaviour of copper determined by cyclic voltammetry (Qin et al. 2017). a) The effect of pH and $[Cl^-]$ on the A/P behaviour in $0.01 \text{ mol/L SO}_4^{2-}$ at $25 \text{ }^\circ\text{C}$. b) A/P map showing regions of active and passive behaviour in $0.01 \text{ mol/L SO}_4^{2-}$ solution at various temperatures.

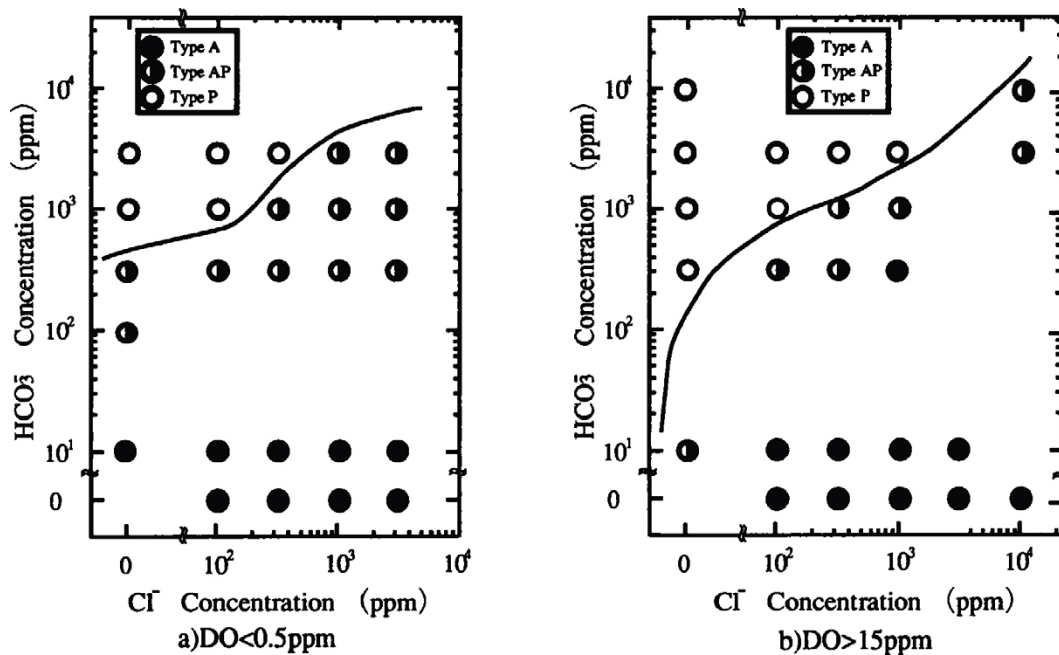


Figure 2-4. Effect of bicarbonate and chloride concentrations on the A/P behaviour of copper in a) deaerated and b) oxidic solutions at 30 °C (Imai et al. 1996).

2.2 Pit initiation and growth

Passive surfaces can be subject to localised film breakdown leading to pit initiation and growth. The electrochemical consequences of film breakdown and pitting can be seen from the passive cyclic voltammogram (CV) in Figure 2-1. Film breakdown is characterized by the rapid increase in current during the forward (anodic-going) potential scan, at a potential (E_B) of approximately $-0.12 V_{SCE}$. The breakdown potential E_B is typically defined by the intersection of tangents to the passive and rapidly rising parts of the CV. With increasing polarisation, the current increases as the newly initiated pit propagates. If the potential scan is reversed, the current measured on the reverse scan exceeds that on the forward scan (referred to as positive current hysteresis) and is a characteristic of pit growth. At some point, however, the current decreases to that measured during the forward scan indicating that the propagating pit has ceased to grow and that the surface has repassivated. There are several methods for defining the characteristic repassivation potential (E_{RP}), but those commonly used are the potential at which the forward and reverse scans cross one another or the potential for a specific current density on the reverse scan.

The precise meaning of E_B and E_{RP} have been long debated (Thompson and Syrett 1992, Soltis 2015 and references therein). The breakdown potential is the potential above which stable pits can grow, since metastable pitting (pit initiation events, but no growth) are commonly observed at potentials more-negative than E_B . The repassivation potential is more unequivocally associated with the cessation of pitting and has been regarded as the potential necessary to sustain the critical chemistry within the pit, as well as the minimum potential for metastable pitting. For this reason, E_{RP} has been proposed as a conservative measure of pit initiation for the long-term prediction of localised corrosion (Sridhar and Cragolino 1993).

One of the problems encountered with the use of critical potentials for predicting localised corrosion behaviour is that both E_B and E_{RP} are found to be dependent on the method used to measure them (Soltis 2015). Thus, in addition to the effect of the environmental parameters of interest, E_B may also be a function of potential scan rate, surface finish, and electrode pre-treatment. The repassivation potential can be a function of the pit depth, i.e., the scan rate or the time at which the pit is allowed to grow potentiostatically. In addition to these dependences on the measurement method, the value of E_B especially is variable due to the stochastic nature of pit initiation. Figure 2-5 shows the cumulative distributions of E_B and E_{RP} values measured from 21 identical experiments, showing the inherent variation in the critical potentials in the absence of effects due to different measurement techniques.

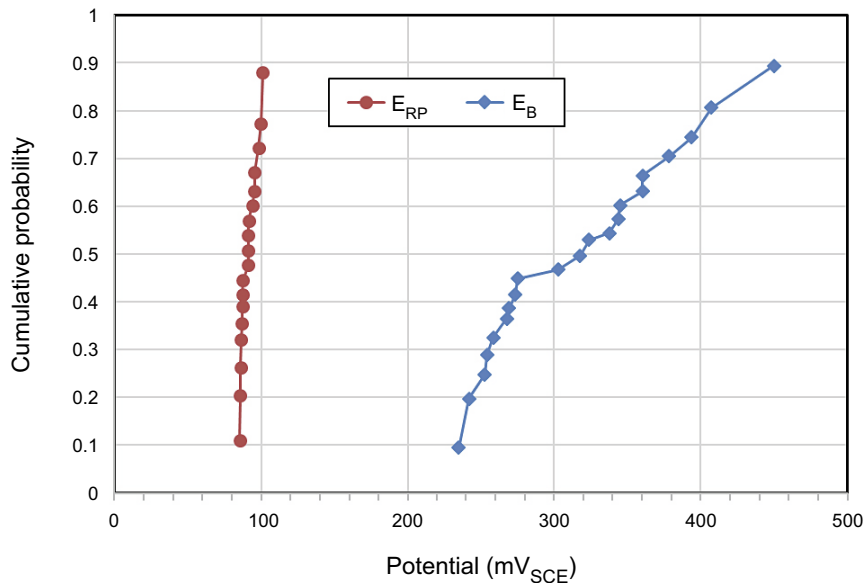


Figure 2-5. Cumulative distribution plots for replicate measurements of the repassivation and breakdown potentials of copper in a synthetic drinking water solution at pH 9.5 (based on the data of Cong et al. 2009).

2.3 Criteria for pit initiation and propagation

The following criteria for the initiation and growth of pits are used for this study:

1. The surface must be passive for pit initiation to be possible.
2. Pit initiation occurs if the corrosion potential E_{CORR} exceeds the value of the film breakdown potential E_B .
3. Once a pit has initiated, propagation occurs as long as E_{CORR} exceeds the value of the repassivation potential E_{RP} .

Given the different interpretations and sources of variability for E_B and E_{RP} described above, the choice of these criteria has certain implications for the model predictions. The use of the criterion $E_{CORR} > E_B$ for pit initiation implies that the model predicts the initiation (and subsequent growth) of stable, as opposed to metastable, pits. Furthermore, for both pit initiation and repassivation, it is implicitly assumed that the E_B and E_{RP} data used in the analysis account for the different sources of variability described above.

As will be described in more detail below, the values of the two critical potentials and of E_{CORR} are time dependent as a result of the evolution of the repository environment. Furthermore, the active or passive nature of the canister surface also evolves with time as the environment changes.

3 Input data

The background information described in the previous section provides the basis for the probabilistic pitting model and defines the types of input data that are required. The availability of such data is described in this section. As described in Section 2.3, pit initiation and stability will be judged based on the requirement for passivity (Section 3.1.1) and on the comparison of E_{CORR} (Section 3.1.4) with the critical potentials E_B and E_{RP} (Section 3.1.2). If pit propagation is possible, the extent of growth will be based on empirical pit growth kinetics (Section 3.1.3).

Information is also required about the relevant environmental conditions, including: canister temperature (Section 3.2.1), pore-water pH (Section 3.2.2), chloride concentration (Section 3.2.3), concentrations of sulphate and bicarbonate (Section 3.2.4), and, in order to estimate the value of E_{CORR} , of the dissolved O_2 concentration (Section 3.2.5). In addition to the possible time-dependent evolution of these parameters, it is also necessary to take into account the uncertainty and their spatial variability within the repository.

3.1 Corrosion and electrochemical data

3.1.1 Conditions for active or passive behaviour

Although a number of investigators have studied the conditions for the passivation of copper (Cong et al. 2009, Imai et al. 1996, King et al. 2010 and references therein), the most extensive study was that performed by the University of Western Ontario (UWO) for the Nuclear Waste Management Organization, the results of which are summarised by Qin et al. (2017). The database from that study was made available by NWMO for the current project. In contrast to the wide range of environmental conditions in the UWO study, the passivation studies of Cong et al. (2009) and Imai et al. (1996) were limited to a single temperature. In addition, the pH was neither systematically varied nor indicated in the study of Imai et al. (1996).

Qin et al. (2017) considered the effects of temperature, pH, and of the concentrations of Cl^- , SO_4^{2-} , and HCO_3^- on the passivation of copper. Cyclic voltammetry was used to distinguish active and passive behaviour (Figure 2-1) resulting in A/P maps of the type shown in Figure 2-3. A total of 722 voltammograms were recorded, of which 458 were categorised as displaying active behaviour and 264 passive behaviour. The ranges of environmental parameters considered are given in Table 3-1. Single salt as well as binary and ternary salt mixtures were considered, with the compositions and concentrations selected to represent those expected in the bentonite pore water during the early saturated phase in the repository. The temperature range was selected to cover the range expected during the thermal transient, and the pH was varied from near-neutral to a value sufficiently high to induce passivation in a given salt solution.

Table 3-1. Ranges of environmental parameters considered in the active-passive cyclic voltammetry studies of Qin et al. (2017).

System	[Cl ⁻] (mol/L)	[SO ₄ ²⁻] (mol/L)	[HCO ₃ ⁻ /CO ₃ ²⁻] (mol/L)
Single salt	0.001–5.0	-	-
	-	0.001–0.1	-
Binary salt mixtures	0.001–5.0	0.01	-
	-	0.01	0.0001–0.01
	0.01	0.001–0.1	-
	0.01	-	0.0001–0.01
	0.1	-	0.0001–0.0005
Ternary salt mixture	0.1	0.01	0.0001–0.0005
Temperature (°C)	25–80		
pH	7.0–12.4		

In summary, the effects of temperature and pH on the passivation of copper were determined for the following combinations of anions:

- Single ion effects of varying concentrations of Cl^- or SO_4^{2-}
- Binary mixtures of
 - Cl^- (variable) and SO_4^{2-}
 - SO_4^{2-} and $\text{HCO}_3^-/\text{CO}_3^{2-}$ (variable)
 - Cl^- and SO_4^{2-} (variable)
 - Cl^- (0.01 mol/L) and $\text{HCO}_3^-/\text{CO}_3^{2-}$ (variable)
 - Cl^- (0.1 mol/L) and $\text{HCO}_3^-/\text{CO}_3^{2-}$ (variable)
- A ternary salt mixture of Cl^- , SO_4^{2-} , and $\text{HCO}_3^-/\text{CO}_3^{2-}$ (variable)

Either active or passive behaviour was assigned for each of the 722 experiments. Expert judgement was used to decide whether the behaviour was A or P in the cases where the shape of the voltammogram was not as clearly distinct as the examples shown in Figure 2-1. Although there is also uncertainty associated with the precise position of the A/P boundaries, such as those illustrated in Figure 2-3(b), this source of uncertainty was addressed using the classification methods described in Section 4.3.1.

3.1.2 Critical potentials

There are a number of sources of critical potentials for copper in relevant environments (King and Lilja 2014, King et al. 2010). However, many of these studies were of limited scope with a relatively small number of E_B and E_{RP} values having been reported. The two most-extensive studies are those of Cong et al. (2009) and Qin et al. (2017).

Cong et al. (2009) investigated the pitting of copper in potable water and determined critical potentials (as well as conditions for passivity) as a function of pH and Cl^- , SO_4^{2-} , and $\text{HCO}_3^-/\text{CO}_3^{2-}$ concentrations. The effect of $\text{Cl}^- - \text{SO}_4^{2-}$ and $\text{Cl}^- - \text{HCO}_3^-/\text{CO}_3^{2-}$ interactions were considered at pH 8.3 and 9.5. The ranges of anion concentration were similar to those used by Qin et al. (2017) (Table 3-1), except for a maximum $[\text{Cl}^-]$ of 0.5 mol/L. All experiments were conducted at a single temperature of 25 °C (room temperature).

Linear regression was used to develop relationships between E_B (referred to as E_{Pit} by Cong et al.) and E_{RP} , namely:

$$E_B = 1.11 + 0.116 \cdot \log[\text{OH}^-] + 0.197 \cdot \log[\text{HCO}_3^-] - 0.130 \cdot \log([\text{SO}_4^{2-}] + [\text{Cl}^-]) V_{SCE} \quad (3-1)$$

and

$$E_{RP} = 0.073 + 0.0044 \cdot \log[\text{OH}^-] + 0.0782 \cdot \log[\text{HCO}_3^-] - 0.0593 \cdot \log[\text{SO}_4^{2-}] V_{SCE} \quad (3-2a)$$

$$E_{RP} = -0.0925 + 0.00373 \cdot \log[\text{OH}^-] - 0.0139 \cdot \log[\text{HCO}_3^-] - 0.0566 \cdot \log[\text{Cl}^-] V_{SCE} \quad (3-2b)$$

for $\text{Cl}^- - \text{SO}_4^{2-}$ and $\text{Cl}^- - \text{HCO}_3^-/\text{CO}_3^{2-}$ solutions, respectively. It is interesting to note that, whereas film breakdown (E_B) is a strong function of the environmental conditions, repassivation (E_{RP}) is only weakly dependent. The authors cautioned that the applicability of these relationships outside of the experimental pH range was uncertain.

As shown in Figure 2-5, Cong et al. (2009) also determined the variability in E_B and E_{RP} as a result of the stochastic nature of film breakdown and, to a lesser extent, repassivation. The replicate E_B measurements exhibit a wider spread of values than the E_{RP} data, with standard deviations of the values in the figure of 61 mV and 5.1 mV for E_B and E_{RP} , respectively. This variability does not account for uncertainty or variability due to differences in the methodology used for the measurements since all values were determined using the same technique.

Qin et al. (2017) also determined E_B and E_{RP} values from the subset of cyclic voltammograms that exhibited passive behaviour. Of the 264 voltammograms that displayed passivity, a total of 254 E_B values and 203 E_{RP} values were derived, with not all experiments exhibiting clear breakdown or repassivation phenomena. As discussed in Section 4.3.2, various regression techniques were used to fit these values to the five experimental parameters. These regression fits were conducted within the probabilistic code itself and fits of the types shown in Equations (3-1) and (3-2) were not developed for these data.

3.1.3 Pit growth kinetics

If the initiation of a stable pit is predicted, pit growth is assumed to continue until such time that the pit repassivates, as indicated by a decrease of E_{CORR} to the value of E_{RP} . The model, therefore, will predict a pit growth duration from which an estimate of the depth of penetration can be determined using either an empirical pit growth expression or from the depth of general corrosion and a suitable pitting factor.

Typically, the rate of pit (depth wise) growth is time-dependent, with the pit depth D at a time t given by

$$D = At^n \quad (3-3)$$

where the value of the time exponent n is < 1 and a value of approximately 0.5 is commonly reported. There is some mechanistic basis for a time-dependent pit growth rate, for example, diffusion control or limitation by an iR drop down an ever-deepening pit would both be consistent with a value of $n < 1$.

For the current model, values for the constant A and the time exponent n are taken from the underground corrosion study of Denison and Romanoff (1950). This study involved the burial of copper samples (as well as of a range of ferrous materials, lead, and zinc alloys) in a range of soils in the United States for periods up to 14 years (Romanoff 1989). These data were used to derive a value for the pitting factor in the original study of the corrosion behaviour of copper canisters by the Swedish Corrosion Institute (SKBF/KBS 1978). In their analysis, Denison and Romanoff (1950) combined results for copper-silicon alloys and for tough-pitch and phosphorus-deoxidized coppers into a single database as these materials exhibited similar behaviour. Of the fourteen different soils types to which copper samples were exposed, only 6 were described as having fair or good aeration and three of these soils were acidic. Data for soils with poor or very poor aeration or for acidic soils are excluded here as being unrepresentative of the repository near field during the aerobic phase.

Pit growth expressions were selected from the two of the remaining three soils exhibiting the deepest pits; soil #65 (Chino silt loam, pH 8, good aeration) and soil #66 (Mohave fine gravelly loam, pH 8, fair aeration) (Table 3-2). These two soils exhibit different pit growth kinetics. Although the maximum pit depth after 14 years was higher for soil #65 than that for soil #66 (406 μm and 257 μm , respectively), the short-term rate is higher for soil #66. The constant A represents the pit depth after one year and is higher in soil #66 but decreases more rapidly, as indicated by the smaller value of n . The consequence for the estimation of pit depth in the probabilistic model is that if the duration of the pitting event is relatively short (of the order of a year or so) the use of data for soil #66 will produce the deepest pits, whereas the data for soil #65 will produce deeper pits if the growth period is of the order of 5 years or greater (Figure 3-1).

Table 3-2. Pit growth parameters for copper for selected soils from the study of Denison and Romanoff (1950).

Soil designation	Soil pH and degree of aeration	Constant A ($\mu\text{m} \cdot \text{yr}^n$)	Time exponent (n)
#65 Chino silt loam	pH 8, good	89.2 ± 46^a	0.57 ± 0.26^a
#66 Mohave fine gravelly loam	pH 8, fair	188 ± 41^a	0.12 ± 0.11^a

^a The \pm is based on the standard error given in the original study and is assumed here to represent one standard deviation for a normal distribution.

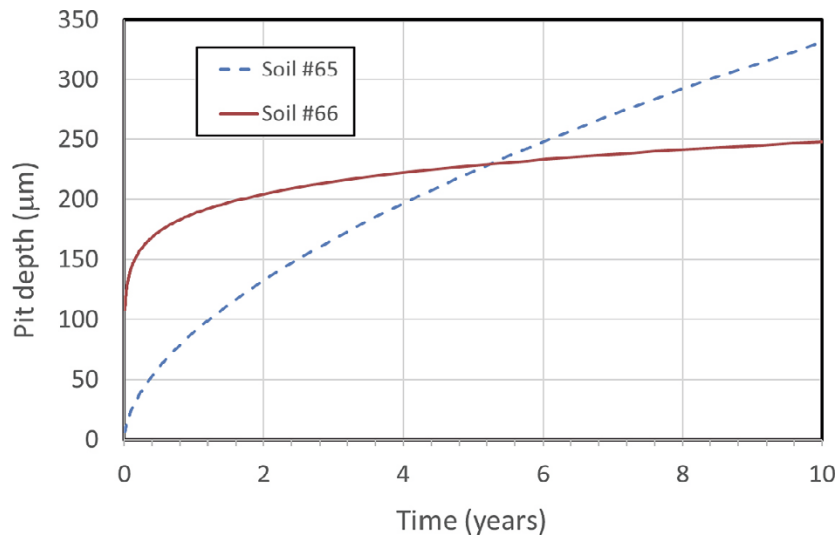


Figure 3-1. Time dependence of the pit depth for soils #65 and #66 from Table 3-2.

An alternative approach to estimating the extent of pit propagation is through the use of a “pitting factor”, defined as the ratio of the depth of maximum penetration to the mean depth of corrosion. Thus, a pitting factor of one indicates uniform corrosion. Based on observations from large-scale *in situ* canisters tests at the Äspö HRL, the rate of general corrosion under aerobic conditions is of the order of 1 μm/yr (Johansson et al. 2019). Various values have been defined for the pitting factor for copper canisters (King et al. 2010), ranging from a highly conservative value of 25 to a value of 5 that is considered more reasonable and which is adopted here. Thus, for the period of pit propagation, the maximum rate of propagation is taken to be 5 μm/yr.

3.1.4 Corrosion potential

The value of the E_{CORR} is used to determine (a) if pit initiation is possible and (b) whether repassivation of an actively growing pit occurs. In general, the value of E_{CORR} is expected to be a function of a number of environmental variables and to vary with time as the repository environment evolves. Detailed reactive-transport models have been developed to predict the evolution of the corrosion behaviour and of E_{CORR} of copper canisters under aerobic conditions (King et al. 2008, 2011a, b). However, the incorporation of such computational complex calculations within a probabilistic model that will be run tens of thousands of times is not feasible and a simpler expression for E_{CORR} is required.

King et al. (1995) described a mixed-potential model for copper in O_2 -containing Cl^- solutions that was applicable to a wide range of mass-transport conditions. Copper anodically dissolves as $Cu(I)$ in the form of $CuCl_2^-$ complex ions (with $CuCl_3^{2-}$ species increasingly prevalent at $[Cl^-]$ greater than approximately 1 mol/L, Puigdomenech and Taxén 2000), supported by the cathodic reduction of O_2 . For the low rates of mass transport characteristic of saturated buffer material, E_{CORR} is a function of temperature and of the concentrations of dissolved O_2 and Cl^- and is given by (King et al. 1995)

$$E_{corr} = E_a^0 + \frac{2.3RT}{F} \log \left[\frac{k_{-2}}{k_a[Cl]^{-2}} \frac{n_c D_{O_2} [O_2]}{n_a D_{CuCl_2^-}} \right] \quad (3-4)$$

where the various symbols are defined in Table 3-3. Because of the dependence of E_{CORR} on $[O_2]$, we also need to define the evolution of the dissolved oxygen concentration in the repository (Section 3.2.5).

3.1.5 Applicability of electrochemical and pit growth data

In the absence of electrochemical data measured in the presence of compacted bentonite, parameter values determined in bulk solution are used and are implicitly assumed to be valid for the compacted system. Compacted bentonite could have a number of effects on the interfacial electrochemical behaviour, including:

- The exertion of a swelling pressure and the resulting effect on film properties such as the porosity and the presence of defects (cracks).
- Buffering of the interfacial pH by equilibration with the bentonite pore water.
- Possible cathodic limitation on the extent of pit growth and the probability of repassivation.

It is uncertain whether the application of a buffer swelling pressure of up to 10 MPa will have an effect on the properties of the film formed on copper. It could be argued that such films will be more compact and less porous and, therefore, more protective than similar films formed in bulk solution. However, porous corrosion product deposits are formed in confined environments, such as in crevices between steam generator tubes and the tube support plate. It is considered unlikely that the presence of swelling bentonite would result in more-porous, less-protective surface films, so that the use of data from bulk-solution studies can be considered to be conservative.

The bentonite itself and accessory minerals such as calcium carbonate in the bentonite will buffer the pore-water pH, which is one of the key parameters determining active/passive behaviour. This effect of bentonite is accounted for in the current model since the distribution of pH values is selected to reflect that in the buffer pore water (see Section 3.2.2).

Saturated bentonite will also limit the mass transport of species towards and away from the canister surface, especially the supply of O₂ during the aerobic phase. Values for the breakdown and repassivation potentials are typically determined potentiodynamically, with the potential controlled by a potentiostat. Under these conditions, the apparatus delivers as much current as required by the pitting system and there is no cathodic limitation of pit growth. This absence of cathodic limitation during the electrochemical measurements will primarily affect the repassivation process, since E_{RP} depends to some extent on the extent of pit propagation between the breakdown event and the point on the reverse scan at which repassivation occurs. Therefore, it is likely that a pit initiated under conditions of cathodic limitation would repassivate at a more-positive potential than a pit initiated potentiodynamically. Thus, the current analysis will tend to overestimate the duration of the pit growth phase.

The pit growth kinetics in Section 3.1.3 were determined in near-surface aerobic soils described as exhibiting fair or good aeration. This description implies that the rate of O₂ supply to the buried samples was relatively rapid and that the use of the resulting pit growth expressions for a canister surrounded by low-permeability compacted buffer is conservative.

Equation (3-4) for the prediction of E_{CORR} does account for the low rates of diffusion of species towards and away from the corroding copper surface and was, in part, developed on the basis of corrosion potentials measured in compacted buffer. However, the expression is based on the assumption of an active surface. This assumption is valid for 90 % of the cases considered (see below) for which the sampled values of T, pH, [Cl⁻], [HCO₃⁻], and [SO₄²⁻] are consistent with active conditions. However, for the approximately 10 % of cases in which passivation is predicted, Equation (3-4) will tend to predict a value more-negative than the E_{CORR} of a passive surface. As a consequence, film breakdown is less likely to be predicted to occur in the model and, where pit initiation does occur, repassivation will tend to occur sooner. Thus, the duration of pit growth will be under-estimated in the model. However, as shown in Figure 3-1, the pit-growth kinetics tend towards a maximum pit depth (especially for soil #66) so that the under-estimation of the duration of the pit-growth phase, while non-conservative, should not result in too large an error in the predicted pit depth. Furthermore, the under-estimation of the duration of pit growth for passive systems based on the use of Equation (3-4) is offset by the over-estimation of the pitting period because of the use of potentials determined in the absence of possible cathodic limitations (see discussion above).

Table 3-3. Parameter definitions and values for Equation (3-4), from King et al. (1995). The electrochemical rate constants are further specified in King et al. (2008).

Symbol	Parameter description	Value
T	Temperature (K)	Variable
F	Faraday constant	96487 C/mol
R	Gas constant	8.314 J K ⁻¹ mol ⁻¹
n _a	Number of electrons in the anodic reaction	1
n _c	Number of electrons in the cathodic reaction	4
E _a ⁰	Standard potential for the anodic reaction	-0.105 V _{SCE}
k _a	Electrochemical rate constant	3.3 × 10 ⁻⁴ dm ⁴ mol ⁻¹ s ⁻¹
k ₋₂	Electrochemical rate constant	1.42 × 10 ⁻³ dm ³ s ⁻¹
D _{O2}	Oxygen diffusion coefficient in solution*	1.7 × 10 ⁻⁵ cm ² s ⁻¹
D _{CuCl2-}	CuCl ₂ ⁻ diffusion coefficient in solution*	5.5 × 10 ⁻⁶ cm ² s ⁻¹

* The ratio of the diffusion coefficients of O₂ and CuCl₂⁻ in Equation (3-4) is assumed to be the same in saturated bentonite as in bulk solution.

3.2 Environmental data

The corrosion behaviour of the canister is determined by the nature of the environment at the canister surface and how it evolves with time. For much of the time period of interest here, the pore-water composition is largely determined by the type of bentonite used and the associated content of accessory minerals. The composition of the ground water is of secondary importance, except for the pore-water [Cl⁻] where the halite content of the bentonite is low and the influence of the ground water will be greater. The concentration of anions in the interlayer space in the bentonite will be lower than in the external source (ground water) due to osmotic effects (Birgersson and Karnland 2009). Furthermore, it is reasonable to assume that the copper surface acts as an external source rather than as a part of the interlayer. Therefore, the anion concentrations at the canister surface, as well as the pH, are assumed to be those associated with an external source in equilibrium with the bentonite.

3.2.1 Canister temperature

Canister surface temperatures were calculated for each of the 6916 deposition holes at the Forsmark site using the thermal code described by Hedin (2004). Calculations were performed using thermal properties appropriate for saturated buffer and were carried out to a maximum time of 10 000 years. For the probabilistic model simulations, temperatures were interpolated using a piecewise cubic method from those predicted by the thermal model to correspond to the time increments used for the Monte Carlo simulations. For times less than 0.01 years (the earliest output from the thermal model), the temperature was assumed to be constant at this initial value.

For each Monte Carlo realization, the 6916 temperature profiles were sampled using a discrete uniform distribution with no consideration of duplicate selections. A representative set of 100 temperature profiles generated within the probabilistic model are shown in Figure 3-2. The use of 100 realizations is convenient as the data is not too dense for the purpose of visualization. A discrete temperature profile was selected from the total 6916 deposition holes for each of the one million realizations used for the full probabilistic analysis.

The use of thermal properties characteristic of saturated buffer results in lower peak temperatures than would be the case for unsaturated buffer. However, the use of lower temperatures is considered conservative since lower temperatures promote passivation of copper (Qin et al. 2017) and, hence, a higher probability of pit initiation. On the other hand, the thermal calculations were conducted on the assumption of the maximum residual power of 1 700 W per canister, and some canisters will have a lower thermal loading and will be correspondingly cooler.

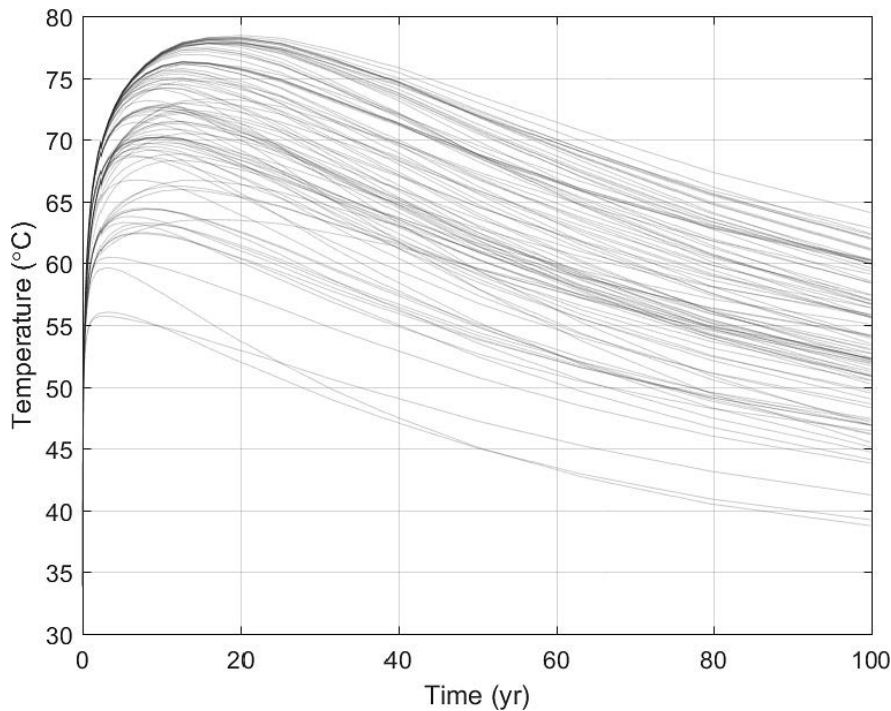


Figure 3-2. Temperature data for a 100 year simulation with 100 Monte Carlo realizations.

3.2.2 Pore-water pH

There are relatively few measurements of the pH of bentonite pore water, largely because of the difficulty of obtaining a representative sample or of conducting *in situ* measurements. Therefore, the form of the pH distribution for the probabilistic model was defined based on expert judgement, informed by the few pH measurements that have been performed and from the results of pore-water modelling studies.

The pH in the bentonite will be determined by calcite equilibrium, ion-exchange and acid-base reactions with the montmorillonite surface, and the composition of the groundwater. Muurinen and Carlsson (2008) measured the pH inside compacted bentonite and the results indicate that the pH value is close to that of the external solution, at least at near-neutral conditions.

A triangular distribution of pore-water pH was used, with minimum, peak, and maximum values of 7, 7.4, and 10, respectively (Figure 3-3). The lower value (pH 7) reflects the lowest measured pH values at the Forsmark site, while the maximum value (pH 10) covers measurements from bentonite porewater (Wanner et al. 1992). The value of pH 10 was measured using a slurry of bentonite soda (Na_2CO_3)-treated bentonite and is considered to be unrepresentative of the pore-water pH in compacted commercial bentonite. Nevertheless, it is used here to define the maximum pH value of the triangular distribution in order to be conservative since the high pH will tend to promote passivation of the surface. The peak value is defined as pH 7.4 in order to (i) reflect a typical value for groundwater and (ii) not give the higher pH values too much weight. As part of a sensitivity analysis, the maximum pH was lowered to pH 9, with the minimum and peak pH values unchanged.

The triangular distribution was sampled at random for each Monte Carlo realization which for a simulation with 1 000 000 realizations, resulted in a distribution shown by the histogram in Figure 3-4. The resulting distribution accurately approximates the desired triangular distribution indicating that sufficient samples were taken in the Monte Carlo analysis.

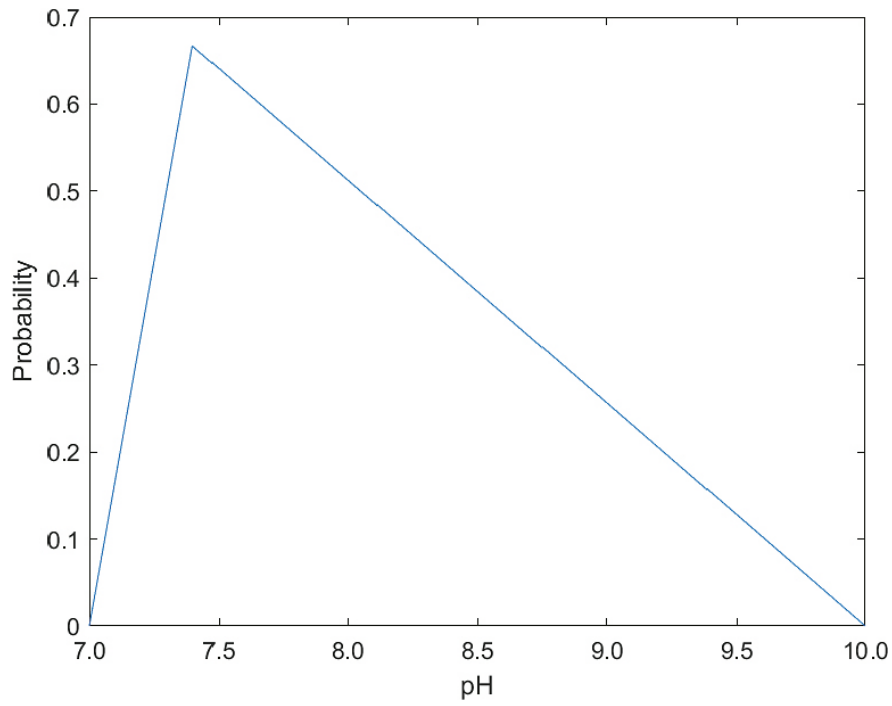


Figure 3-3. Defined pore-water pH triangular distribution with minimum 7, peak 7.4 and maximum 10.

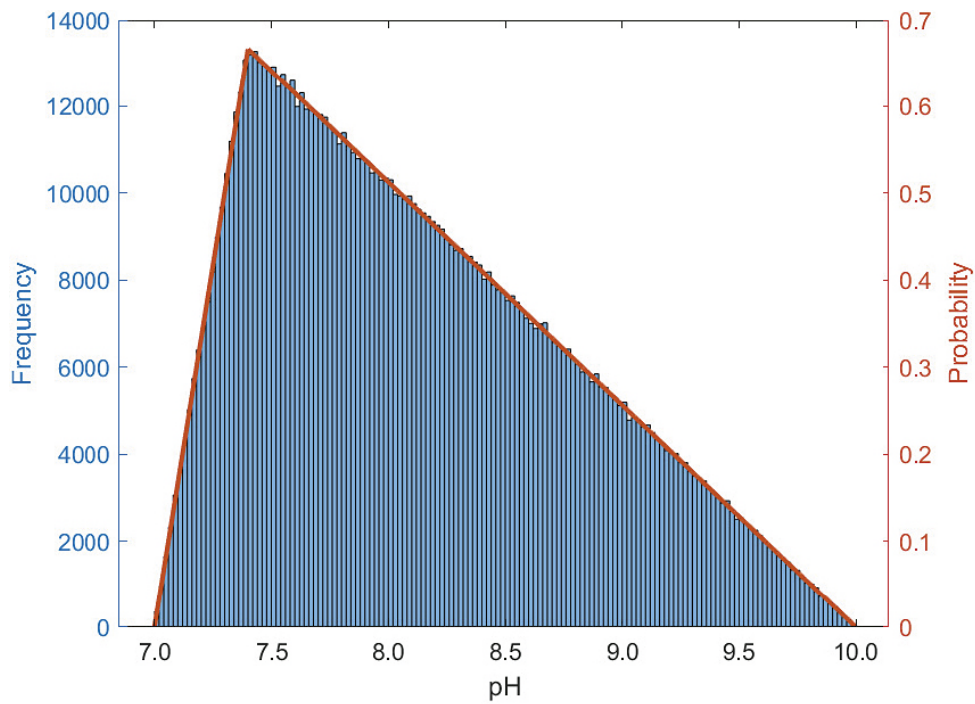


Figure 3-4. Histogram from defined triangular pH distribution for a 1 000 000 realization simulation along with the input triangular probability distribution.

3.2.3 Chloride concentration

The chloride concentration in the bentonite will be determined by the chloride concentration in the groundwater, the initial chloride content in the bentonite, and the degree of saturation. The degree of saturation will depend on the temperature and the supply of water. After full saturation and equilibration, the water in contact with the canister will have the same chloride concentration as the groundwater. However, before equilibrium is reached, which can take a considerable time in some deposition holes, the chloride concentration can take any value between the initial chloride content and the groundwater concentration. The distribution of chloride concentrations is selected based on experimental data from the Prototype Repository (Olsson et al. 2013). This distribution should not be seen as representative for all deposition holes at Forsmark, but more as an illustration of a single possible distribution. However, since the 8-year duration of the Prototype Repository experiment is of the same order as the duration of the aerobic phase modelled here, the distribution of bentonite pore-water chloride concentrations is considered to be representative for the period of interest.

A cumulative distribution function (CDF) of the Prototype data is shown in Figure 3-5 (blue line). In order to smooth out the resulting distribution a monotonic, linearly interpolated, CDF was also generated (Figure 3-5, red line). Finally, the linearly interpolated CDF was randomly sampled for each Monte Carlo realization. For a simulation with 1 000 000 realizations the resulting histogram of Cl^- concentrations is shown in Figure 3-6.

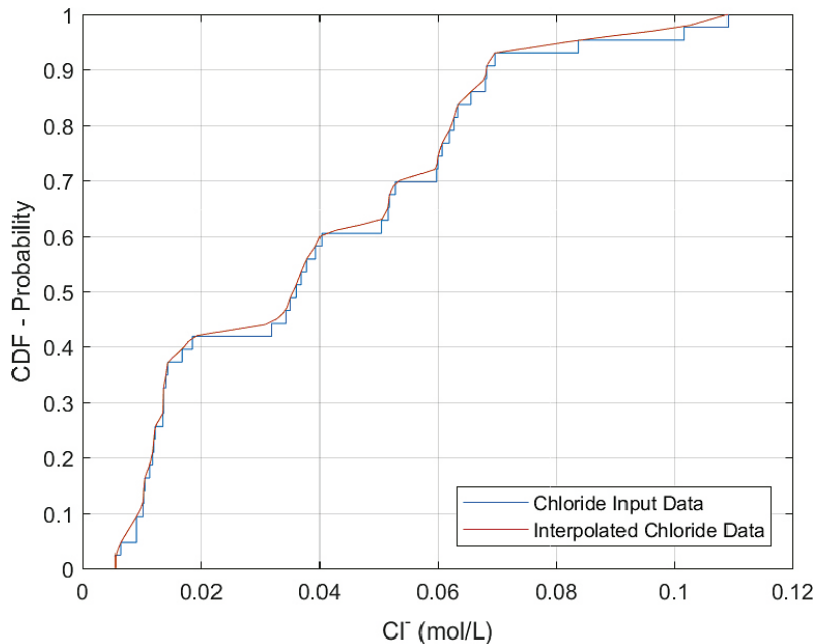


Figure 3-5. Empirical cumulative distribution function (CDF) of pore-water chloride concentrations generated from data from the Prototype experiment (blue) and a linearly interpolated CDF (red).

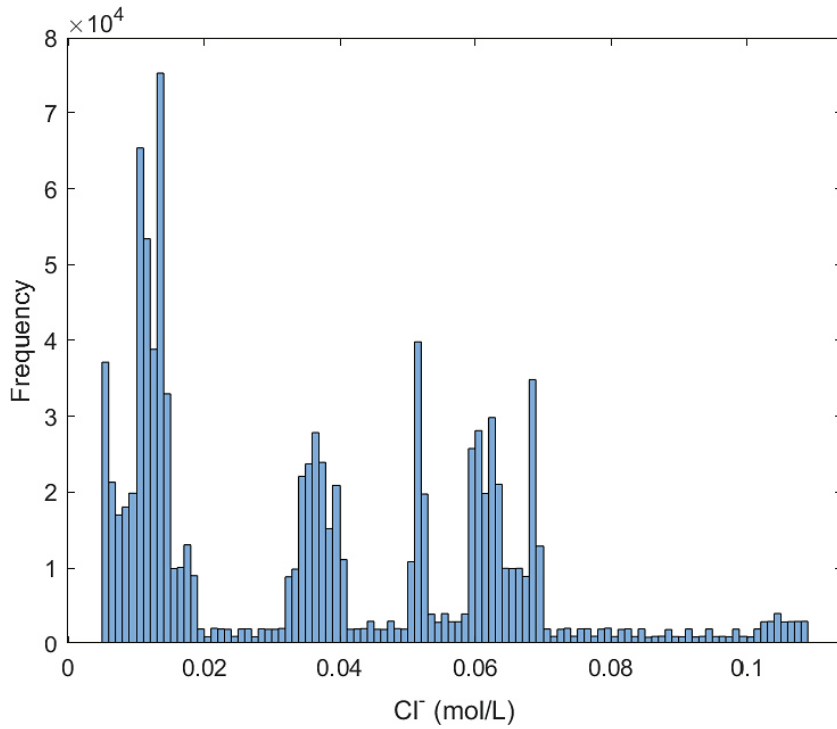


Figure 3-6. Histogram of chloride concentrations for a 1 000 000 realization simulation.

3.2.4 Sulphate and bicarbonate concentrations

The pore-water concentrations of dissolved sulphate and bicarbonate are assumed to be controlled by the solubility of the respective solid phases (King et al. 2017, Sena et al. 2010). A number of the solubility-controlling solid phases exhibit retrograde solubility. The geochemical software PHREEQC version 3.4 was used with the phreeqc.dat thermodynamic database to estimate the equilibrium SO_4^{2-} and HCO_3^- concentrations. Calcite was used as the carbonate controlling phase, with gypsum, anhydrite, basanite, and hemihydrate as possible limiting phases for SO_4^{2-} . The pH was not defined but instead was calculated within the PHREEQC simulation. This may result in a discrepancy between the pH calculated in the solubility calculations and the sampled pH for each realization. However, any such discrepancy is expected to have a relatively minor effect given the overall simplification of the pore-water chemistry in the model.

Polynomials were fitted to the PHREEQC predictions to produce expressions for the temperature dependence of $[\text{SO}_4^{2-}]$ and $[\text{HCO}_3^-]$

$$\begin{aligned}
 [\text{SO}_4^{2-}] = & -1.26901382 \times 10^{-12}(T - 298.15)^6 + 3.97938117 \times 10^{-10}(T - 298.15)^5 \\
 & - 4.74084833 \times 10^{-8}(T - 298.15)^4 + 2.68766739 \times 10^{-6}(T - 298.15)^3 \\
 & - 7.73780299 \times 10^{-5}(T - 298.15)^2 + 1.13158668 \times 10^{-3}(T - 298.15) \\
 & + 7.89962675 \times 10^{-3}
 \end{aligned} \quad (3-5)$$

and

$$\begin{aligned}
 [\text{HCO}_3^-] = & -5.10348861 \times 10^{-10}(T - 298.15)^3 + 1.33530950 \times 10^{-7}(T - 298.15)^2 \\
 & - 1.33530950 \times 10^{-5}(T - 298.15) + 6.33530950 \times 10^{-4}
 \end{aligned} \quad (3-6)$$

and as shown in Figure 3-7.

When the above polynomials are combined with the temperature profiles from Figure 3-2 the resulting sulphate and bicarbonate profiles were produced (Figure 3-8). For each temperature profile that was generated, the corresponding sulphate and bicarbonate concentration profiles were also generated for use in the simulations.

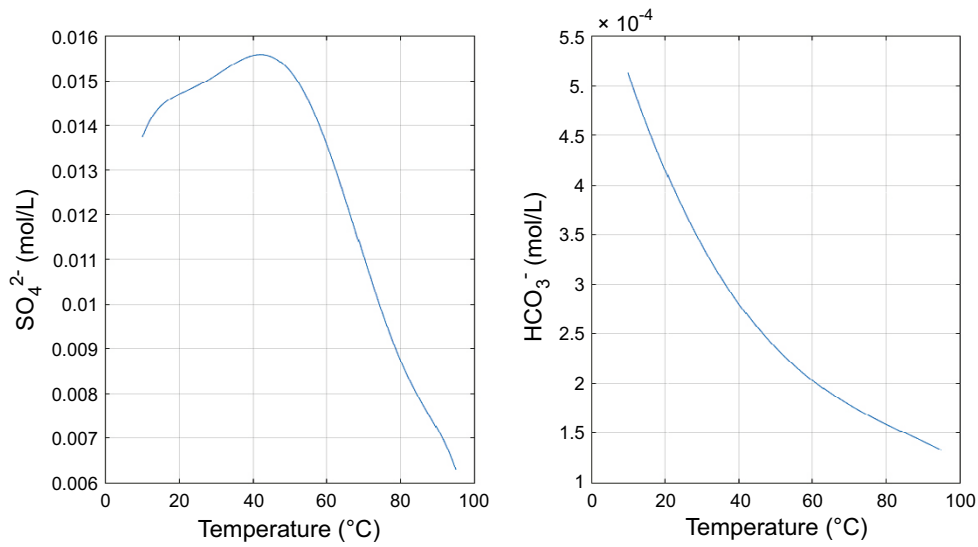


Figure 3-7. Solubility of sulphate and bicarbonate as a function of temperature based on PHREEQC version 3.4 calculations using the phreeqc.dat database.

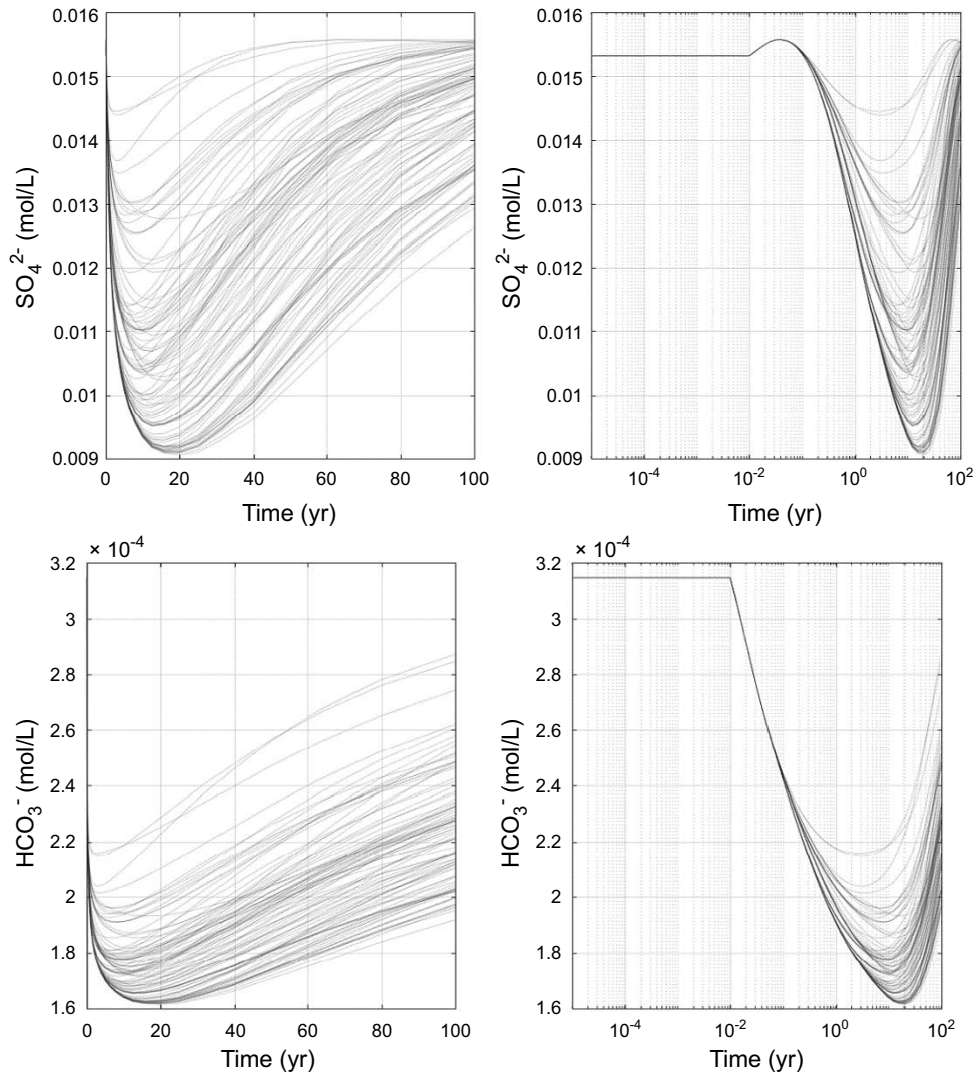


Figure 3-8. Time-dependent concentrations of sulphate (top) and bicarbonate (bottom) for a Monte Carlo simulation with 100 realizations. Results are shown for both linear (left) and log (right) time axes. For times less than 10^{-2} years (the earliest time at which temperatures were calculated using the thermal code), the temperature, and hence the anion concentrations, were assumed to be constant at the initial values.

3.2.5 Oxygen concentration

The time-dependent dissolved O_2 concentration is required to predict the value of E_{CORR} (Equation 3-4). The $[O_2]$ in Equation (3-4) refers to the concentration in the bulk environment, not that at the canister surface (King et al. 1995). As such, it is taken to be represented by the overall evolution of the redox conditions within the repository.

The time-dependence of the O_2 concentration was based on expert judgement. An exponential decrease in oxygen was assumed

$$[O_2] = [O_2]_{init} \cdot e^{-\alpha t} \quad (3-7)$$

where $[O_2]_{init}$ is the initial dissolved oxygen concentration assumed to be 2.4×10^{-4} mol/L, equivalent to that in an aerated saline pore-water solution at 25 °C (Battino et al. 1983). Complete consumption (defined here as 0.1 % remaining) was assumed to occur in between 5 weeks and 5 years. This range of times is representative of the estimated times to establish anoxic conditions in different experimental studies in granitic rock (Puigdomenech et al. 2001) and in full-scale tests (Giroud et al. 2018). The time constant α is thus defined by

$$\alpha = \ln(0.001)/t \quad (3-8)$$

Both log uniform and uniform distributions were considered for α , with lower and upper bounds of 72.08 a^{-1} and 1.382 a^{-1} , corresponding to “ O_2 consumption times” of 5 weeks and 5 years, respectively. For the uniform distribution, α is assumed to be uniformly distributed between the lower and upper bounds. For the log uniform distribution

$$10^{\alpha} \sim U(\log_{10}(\alpha_{upper}), \log_{10}(\alpha_{lower})) \quad (3-9)$$

α is log uniformly distributed between the \log_{10} of the upper and lower bounds.

Comparison between the log uniform (Figure 3-9) and uniform distributions (Figure 3-10) for α indicates that the uniform distribution results in a larger portion of realizations tending towards the lower bound of 5 weeks, considered to be more representative of the majority of experimental observations than the upper bound of 5 years (Giroud et al. 2018, Puigdomenech et al. 2001). Therefore, a uniform distribution was chosen for α for the simulations presented in this report.

We do not attempt to define the duration of the “aerobic phase” during which pitting is possible. The duration of any pit-growth event depends on a number of factors, including: (i) the rate of O_2 consumption, (ii) the time dependence of E_{CORR} (which in turn depends on the temperature, $[O_2]$, and $[Cl^-]$), and (iii) the sampled values of E_B and E_{RP} . Thus, it is possible that in some realizations pitting will continue for longer than 5 years if the rate of O_2 consumption is relatively slow and/or if the value of E_{CORR} is relatively noble compared with the value of E_{RP} .

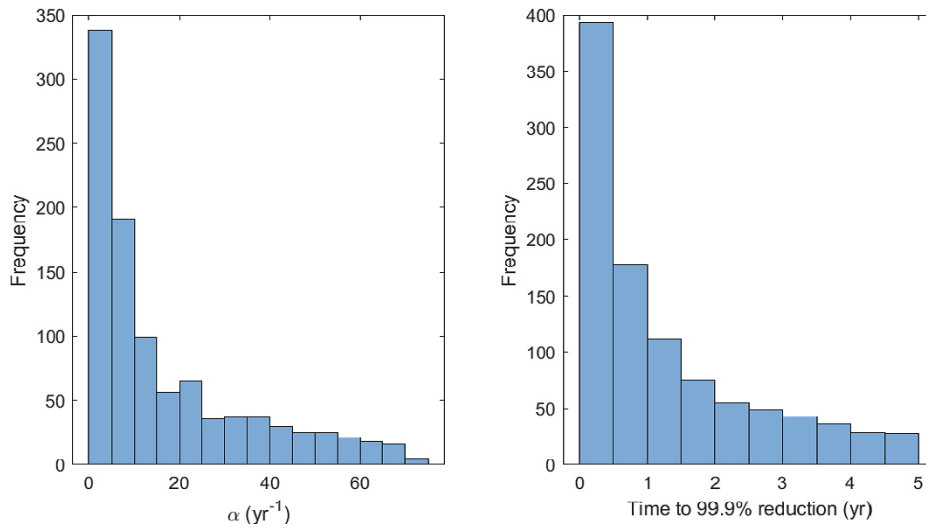


Figure 3-9. Log uniform α distribution (left) with resulting oxygen consumption times (right) for a 1000 realization Monte Carlo simulation.

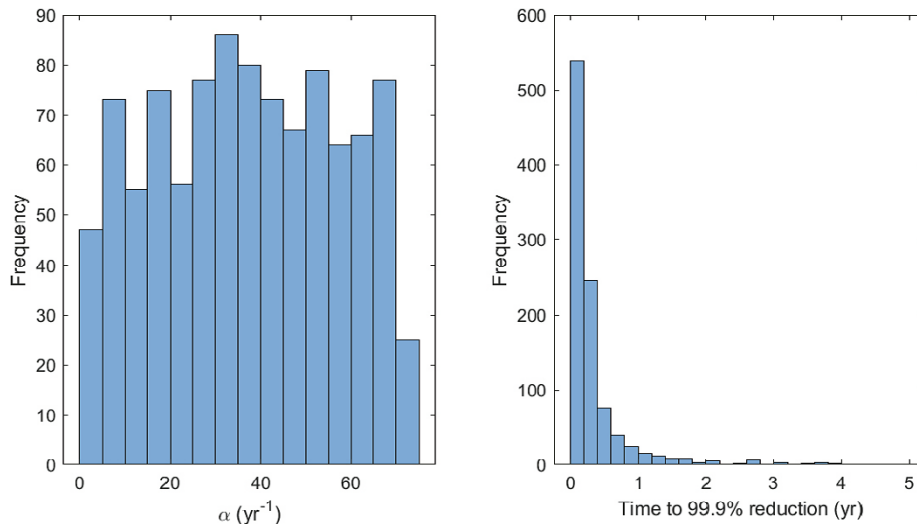


Figure 3-10. Uniform α distribution (left) with resulting oxygen consumption times (right) for a 1000 realization Monte Carlo simulation.

4 Implementation of the probabilistic pitting model

4.1 Structure of the model

The probabilistic model is structured around a series of flow charts that (i) control the overall execution of the code (Figure 4-1), (ii) determine the time-dependent evolution of the environment (Figure 4-2), (iii) determine if pit initiation is possible (Figure 4-3), and (iv) if a pit does initiate, the extent of pit growth (Figure 4-4).

The following terminology is used to describe the execution of the probabilistic model. A *realization* is the term used to represent a single cycle of the overall control flow chart, typically representing an elapsed time of 100 years. A single temperature profile, bentonite pore-water pH and $[\text{Cl}^-]$, and an O_2 -depletion rate constant are selected for each realization. Thus, each realization represents the environmental conditions in a single deposition hole.

A *simulation* represents a complete probabilistic assessment and comprises between 100 and 1 000 000 individual realizations. The smaller number of realizations was used to allow results to be visually presented and to assist in the identification of trends in the predictions, for example, the environmental conditions responsible for passive or active behaviour. The larger number of realizations is used to take into account the variability and uncertainty in the environmental conditions within the deposition holes and within the repository as a whole. A simulation of 1 000 000 realizations represents an average of 145 realizations for each of the 6916 deposition holes. Multiple realizations for a given deposition hole (characterized by a specific temperature profile) account for the variability and uncertainty in the pore-water $[\text{Cl}^-]$ and pH and of the rate of O_2 consumption.

The extent to which the variability and uncertainty in the electrochemical (both the active/passive behaviour and film breakdown and repassivation) and pit growth data is accounted for in a simulation is discussed in Section 4.3.

4.1.1 Control flow chart

The purpose of the Control flow chart (Figure 4-1) is to control the overall execution of the code, including the time-stepping (except during pit propagation), the calling of other modules (flow charts) as needed, and eventually stopping the execution when the realization is over. A single realization represents a series of time increments from emplacement through to 100 years (or some other time, defined by the user). At the beginning of the realization, a single temperature profile is selected at random from the collection of 6916 profiles, one for each deposition hole. In addition, values of the pore-water pH and $[\text{Cl}^-]$ are selected from the corresponding distributions and remain constant for the entire realization. The time dependence of the $[\text{O}_2]$ evolution is also defined by selecting a value for the associated time constant a .

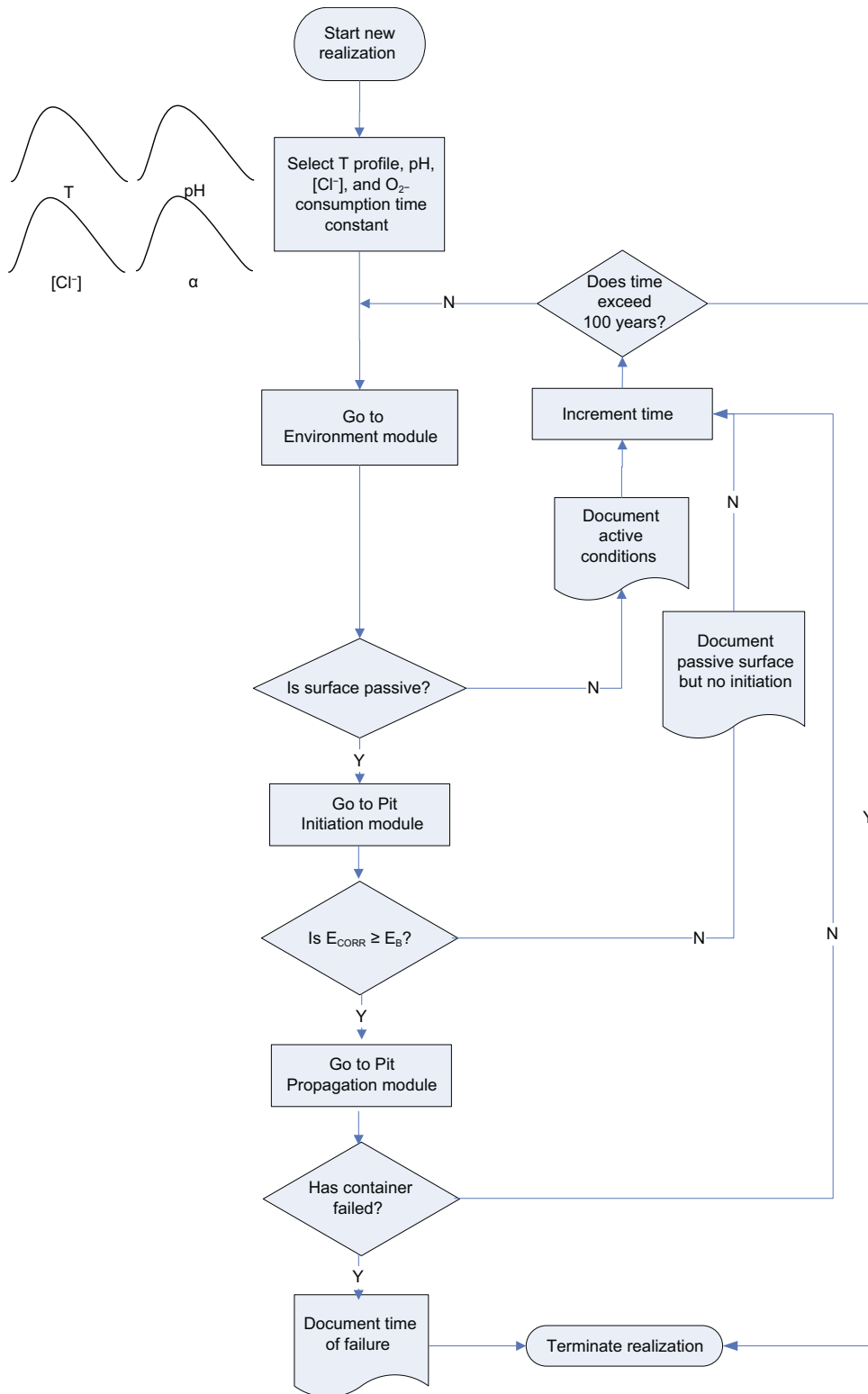


Figure 4-1. Control flow chart for the probabilistic pitting model for aerobic, saturated conditions.

For each time step, the following processes are performed:

- Proceed to the Environment module to determine the composition of the aqueous phase and temperature.
- Determine if the surface is active or passive (performed within the Environment module).
 - If the surface is in the active condition, this observation, and the corresponding time, are documented, the time is incremented, and the above steps are repeated.
 - If the surface is predicted to be passive, however, we proceed to the Pit Initiation module.
- If the surface is passive but the pit initiation criterion is not met, then this observation (and the associated time) are documented, the time is incremented, and the above steps are repeated.
- If a pit is predicted to initiate, we proceed to the Pit Propagation module.
 - The extent of pit propagation is estimated based on the period for which E_{CORR} exceeds E_{RP} and a suitable pit propagation model (e.g., an empirical pit growth rate or a pitting factor).
 - Execution remains within the Pit Propagation module until the surface repassivates.
- Once a pit has stopped propagating, we return to the Control flow chart and check to see if the pit depth equals or exceeds the corrosion allowance.
 - When incrementing the time following a period of pit propagation, the time increment corresponds to the predicted period of pit propagation.
 - By repeating these steps, we allow for the possibility of another period of pit propagation should the various criteria be met.
- Execution stops once either the corrosion allowance is exceeded (and the corresponding “failure time” is recorded) or the time reaches the maximum value specified for the realization (typically 100 years).

4.1.2 Environment flow chart

The two functions of the Environment module (Figure 4-2) are to define the nature of the aqueous phase and to determine whether the surface is active or passive. The canister surface temperature profile, pore-water pH and $[Cl^-]$, and the O_2 -consumption time constant were selected in the prior Control module.

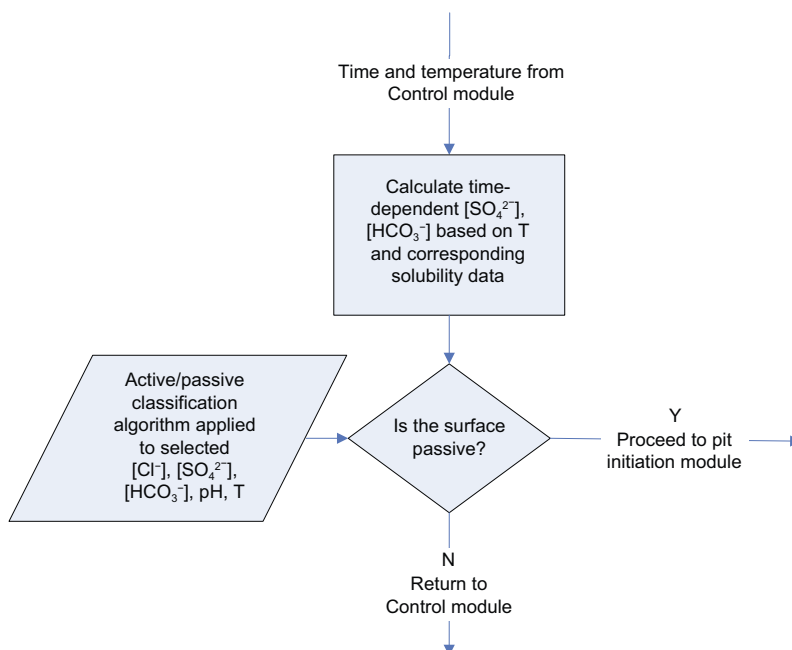


Figure 4-2. Environment flow chart for the probabilistic pitting model for aerobic, saturated conditions.

Each time the Environment module is called, the following steps are performed:

- Define the nature of the aqueous phase.
 - Based on the canister surface temperature for that time step, calculate the $[\text{SO}_4^{2-}]$ and $[\text{HCO}_3^-]$ based on Equations (3-5) and (3-6), respectively.
- Based on the selected T, pH, and $[\text{Cl}^-]$ and the predicted $[\text{SO}_4^{2-}]$ and $[\text{HCO}_3^-]$, determine whether the surface is active or passive based on the data of Qin et al. (2017) and a suitable classification algorithm (Section 4.3.1).
- If the surface is deemed to be passive, then the possibility of pit initiation will be determined in the Pit Initiation module.
- If the surface is deemed to be active, then the simulation is returned to the Overall control module.

4.1.3 Pit Initiation flow chart

The Pit Initiation module (Figure 4-3) is used to determine whether pit initiation occurs based on the criterion that E_{CORR} equals or exceeds the breakdown or pitting potential E_B . The following steps are involved:

- Calculate the $[\text{O}_2]$ based on the elapsed time and the selected O_2 -consumption time constant (Equation 3-7).
- Calculate the value of E_{CORR} based on the defined T, $[\text{Cl}^-]$, and the calculated $[\text{O}_2]$ (Equation 3-4).
- Select a value of E_B based on the T, $[\text{Cl}^-]$, $[\text{SO}_4^{2-}]$, $[\text{HCO}_3^-]$, and pH defined in the Control and Environment modules and the regression analysis of experimental E_B values.
- Determine whether pitting initiates based on the criterion $E_{\text{CORR}} \geq E_B$.
- If pit initiation is predicted to occur, proceed to the Pit Propagation module.
- If pit initiation is predicted not to occur, return to the overall Control module.

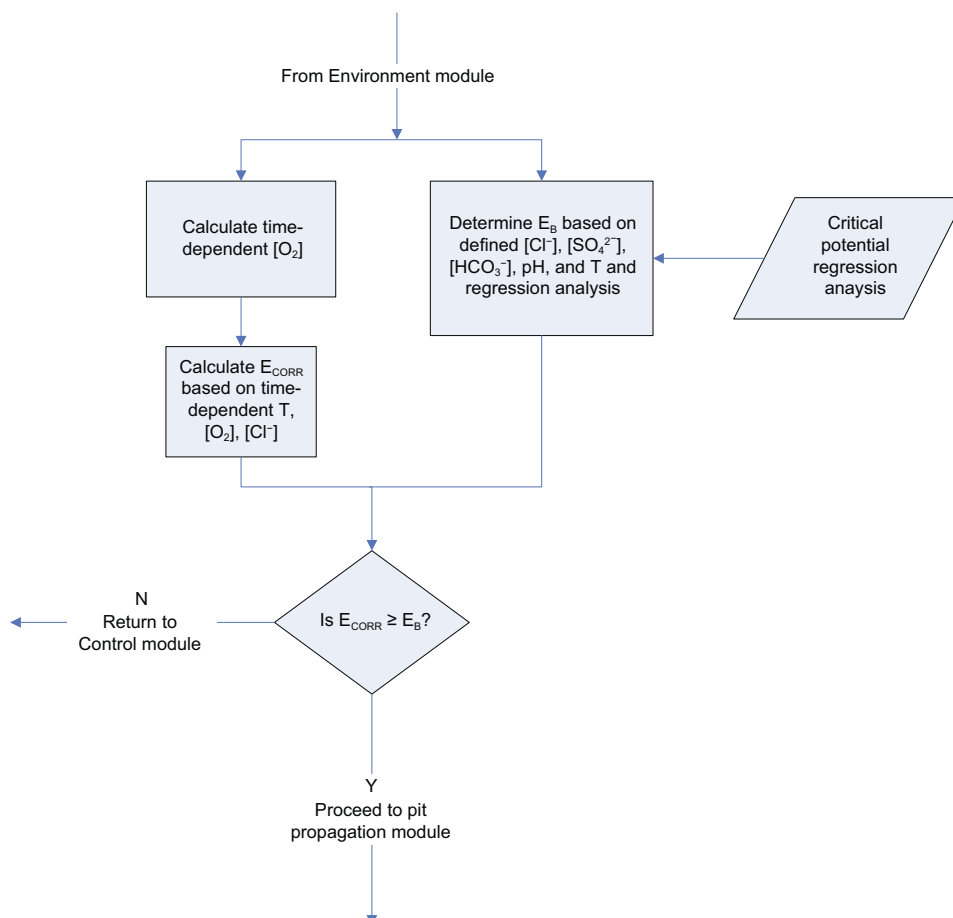


Figure 4-3. Pit initiation flow chart for the probabilistic pitting model for aerobic, saturated conditions.

4.1.4 Pit Propagation flow chart

The Pit Propagation module (Figure 4-4) is used to determine the extent of pit propagation prior to stifling of the pit. The following steps are involved:

- Select a value of the repassivation potential E_{RP} based on the T, $[Cl^-]$, $[SO_4^{2-}]$, $[HCO_3^-]$, and pH defined in the Control and Environment modules and the regression analysis of experimental E_{RP} values.
- Determine if pit growth is possible by comparing the values of E_{CORR} (from the prior Pit Initiation stage) and E_{RP} .
 - This initial step is included as it is possible, depending on the width of the distributions of E_B and E_{RP} , that the selected value of E_{RP} could be more positive than the selected E_B .
 - If $E_{CORR} \leq E_{RP}$, return to the Control module.
- If pit propagation is possible ($E_{CORR} > E_{RP}$), estimate the extent of pit growth during the prescribed time step using an appropriate (user-defined) pit growth expression.
- Document the extent of pit growth during the time step.
- Increment the time and re-calculate the values of E_{CORR} and E_{RP} .
- Re-determine whether pit growth is still possible based on the criterion $E_{CORR} > E_{RP}$.
- If pit growth is possible, estimate the incremental pit growth, and repeat the steps above.
- If $E_{CORR} \leq E_{RP}$, return to the Control module and increment the time based on the total duration of pit growth.

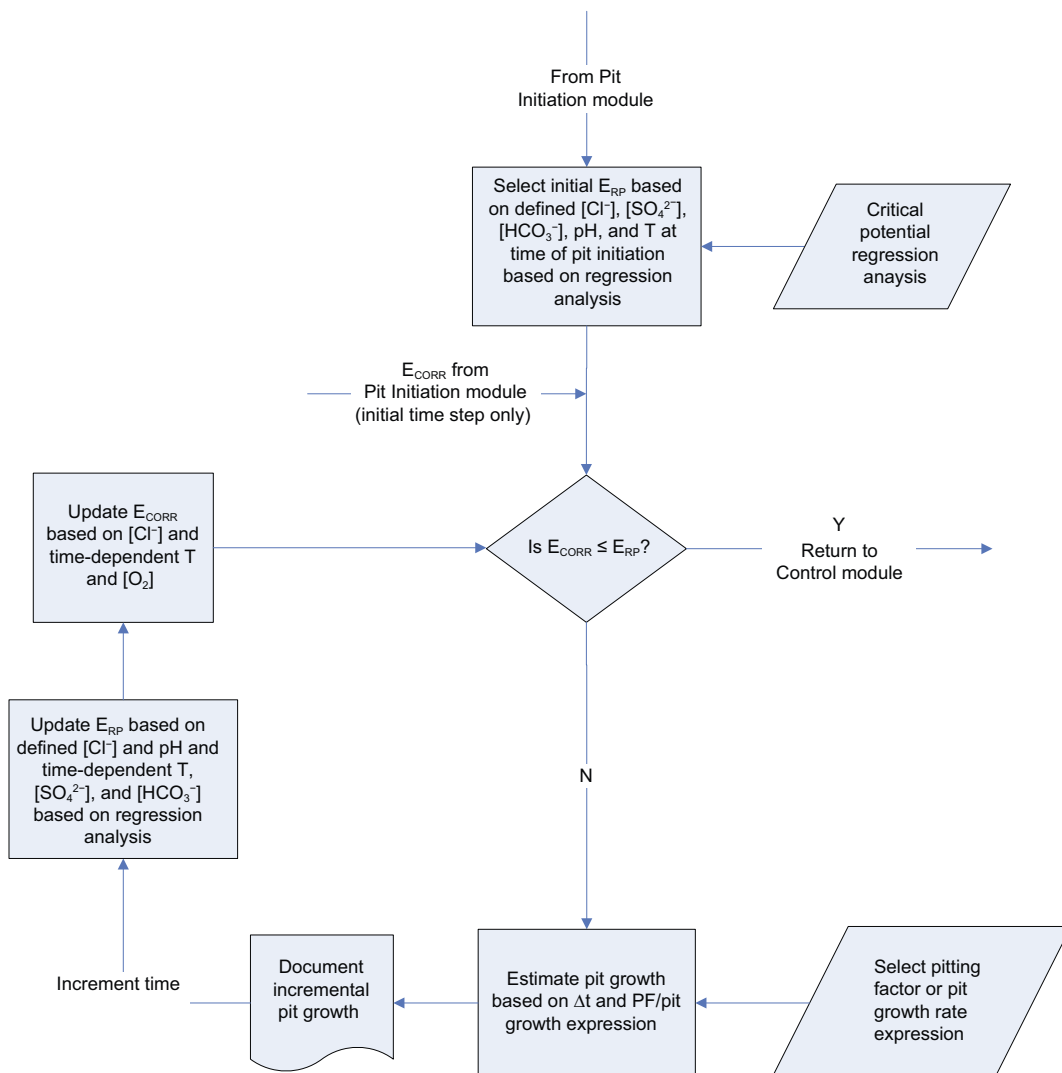


Figure 4-4. Pit propagation flow chart for the probabilistic pitting model for aerobic, saturated conditions.

The Pit Propagation module is responsible for incrementing the time during the period of pit growth and takes over this function from the Control module. During this time, the values of E_{CORR} and E_{RP} are re-calculated at each time step as the temperature, $[O_2]$, $[SO_4^{2-}]$, and $[HCO_3^-]$ evolve. As noted in Section 3.1.2, however, the experimental E_{RP} of Cong et al. (2009) are only weakly dependent on the anion concentrations (Equation 3-2). Once the pit repassivates, execution returns to the Control module and the time is incremented to account for the period of pit growth.

4.2 Software and hardware

Statistical analysis of the experimental data set was conducted using the base software package MathWorks MATLAB with the addition of the Statistics and Machine Learning Toolbox. MathWorks MATLAB is a commercial numerical computing tool for mathematical calculations, such as matrix manipulations, linear algebra and statistical analysis (among many other applications). The MATLAB software and associated toolboxes provide a platform of mathematical calculations based on the well-accepted mathematical libraries known as LAPACK and other algorithms from the literature. They allow users to develop their own calculation programs within this platform. MATLAB has been under development since 1984 by MathWorks. It is widely used in industry and academia. It is being actively developed and maintained.

Development was completed using MATLAB version R2018b (MathWorks 2018). Built-in MATLAB mathematical algorithms were used for most calculations, for example data input, vector operations, interpolation, random number generation, data output and data visualization. The statistical methods also used built-in functions including creating probability distribution and cumulative distribution functions and developing machine learning algorithms for classification and regression.

The software was developed and run on a workstation computer with Intel Xeon processors (total of 20 cores) and 192 GB of error catching and correcting (ECC) memory. Wherever possible the code was developed to be parallel by design, using built-in MATLAB vector operations and running the Monte Carlo (MC) realizations in parallel. When running the largest models (1 million MC realizations) the simulation can take 10 days, using the 20 cores, and take up to 64 GB of memory. For rapid development of the model, smaller simulations can be run in minutes to hours.

4.3 Model development

The model underwent several stages of development using different classification and regression methods, pit growth expressions, and time discretizations. These development steps are described in this section, with the final form of the model used for the full probabilistic simulations described in Section 4.4.

4.3.1 Classification methods for the active-passive data

The input data for the determination of active or passive conditions comprises a set of 722 experimental measurements as a function of different values of T, pH, $[Cl^-]$, $[SO_4^{2-}]$, and $[HCO_3^-]$ (Section 3.1.1). During the execution of the code, a new set of T, pH, $[Cl^-]$, $[SO_4^{2-}]$, and $[HCO_3^-]$ values are defined at each time step and a method is needed of interpolating the experimental data to determine whether the canister surface is active or passive. Since the desired outcome is binary (active or passive), the use of classification methods to determine membership within a given group is appropriate.

Three different classification methods were investigated using the Classification Learner application within MATLAB:

- Unweighted k-nearest neighbour (k-NN) algorithm, using various numbers of nearest neighbours.
- Weighted k-NN.
- Bootstrap aggregating decision tree (Bagging).

In addition, verified and non-verified versions of the weighted k-NN and Bagging methods were investigated, with the “goodness-of-fit” assessed through the use of confusion matrices. These are tables used to describe the accuracy of a classification algorithm. For example, it lists the number of false positives and false negatives along with respective true responses.

In order to condition the input data and minimize error due to large deviations in the relative magnitude between data (for example, the HCO_3^- concentrations are on the order of 10^{-4} while the temperature data are on the order of 10^1 – 10^2), the data were normalized by dividing each value by the standard deviation of the respective data set. These normalized data were used to test all three classification methods.

The k-NN algorithm is a method used for data classification where k represents the number of nearest neighbours used when training on the data. The algorithm calculates the distance between the known data (1 through k nearest neighbours) and a given new data point to determine class membership. In the localised corrosion model, class membership was defined as either active (A) or passive (P). Sensitivity to the number of nearest neighbours was investigated for k values of 1 through 12. In the initial trials, an unweighted k-NN method was used in which all k neighbours have an equal influence on the membership of the new data point regardless of their proximity. Thus, it is difficult, *a priori*, to define the optimum value of k. If k is too small then there may be too few experimental data points to determine membership, especially if the new data point lies close to a “boundary” between A and P behaviour. On the other hand, if k is too large, then experimental points determined under quite different environmental conditions than those of the new data point may have undue influence over the membership classification.

Table 4-1 summarises the results of preliminary Monte Carlo simulations to determine the effect of the number of nearest neighbours k for the unweighted k-NN classification method. Each simulation comprised 100 000 realizations and each run consisted of 71 time steps (total of 7.1 million time steps). In general, the simulations predict more active behaviour the larger the number of nearest neighbours used to determine membership. Thus, not only are $\geq 80\%$ of the individual time steps classified as active for $k \geq 10$, but more than half of the realizations predict active conditions for the entire 100-year simulation period with only 2% predicted to be entirely passive. In contrast, for $k \leq 2$, of the order of 80% of the individual time steps and 50% or more of the entire realizations were predicted to be passive. As noted above, however, it is not obvious whether a smaller or larger number of nearest neighbours is to be preferred for an unweighted k-NN approach.

Table 4-1. Effect of number of nearest neighbours (k) on the distribution of active-passive conditions for a Monte Carlo simulation of 100 000 realizations using the unweighted k-NN classification method.

	k = 1	k = 2	k = 4	k = 6	k = 8	k = 10	k = 12
Time steps:							
% active	27	8	25	52	75	80	84
% passive	73	83	75	48	25	20	16
Realizations:							
% active only	17	4	3	12	15	57	54
% passive only	49	72	12	2	2	2	2

Because of this concern over the possible undue influence of distant neighbours on deciding membership of the new data point, two additional classification methods were investigated. First, a weighted k-NN approach was used in which neighbours closer to the new data point have a greater influence in deciding membership than those further away. For the current analysis, a squared inverse weight was applied to the distance between neighbours to determine their relative influence. Second, the use of a “bagged tree” machine learning algorithm was investigated. Bootstrap aggregating, or bagging, is an algorithm used in statistical classification and regression and is designed to reduce variance and minimize overfitting. The bagging method generates many training sets from the original data (with replacement) and is then used in a decision tree method to apply a classification to each new data set and finally averages the result (Breiman 1996). The classification tree, or decision tree, uses properties of an observation to move towards decisions about the value of an observation. The building of the classification tree is done many times, or trained through machine learning, to create a tree with the highest accuracy for a given dataset. In this way, bagging is an ensemble of methods to obtain higher predictive performance.

In general, when training a machine learning algorithm to a defined dataset, overfitting can be a concern. This is analogous to the use of a high-order polynomial to fit a dataset comprising one dependent variable. The fit can be so “good” that the generated polynomial goes through each data point, but this does not necessarily represent a satisfactory fit to the data. Examples of possible overfitting were found with the weighted and non-weighted k-NN approach. Both the weighted k-NN with $k = 10$ and the non-weighted k-NN approach with $k = 1$ resulted in a 100 % accurate model. The accuracy was calculated by re-substituting existing data points into the generated model to determine if the predicted classification was correct. As a result, the model accurately predicted existing data, but it cannot be determined how new datasets would perform. Therefore, a method of cross-validation (or verification) was applied during the training of the weighted k-NN and bagged tree algorithms. Specifically, n-fold cross-validation was used for both the k-NN and boosted tree algorithms where $n = 5$. In n-fold cross-validation, the data set is randomly sampled into n groups and the selected algorithm is trained on n-1 groups with the final group being used as validation for testing the algorithm. This process is then repeated n times with each group being used as the validation dataset in turn. As a result the algorithms are resistant to overfitting as its accuracy is based on testing against new data. The final trained algorithm then uses all data groups to train before any predictions are made by the model.

Two cross-verified and two non-verified simulations were run for both the bagged tree and weighted k-NN ($k = 10$) algorithms. Table 4-2 summarizes the results of the A/P classifications for the eight simulations. There are number of interesting observations from the data, including:

- The bagged tree and weighted k-NN algorithms both predict active conditions approximately 80–90 % of the time.
- The bagged tree classification method predicts a similar percentage of realizations for which active conditions persist for the entire 100-year simulation period, whereas the k-NN approach predicts only approximately half of the realizations are permanently active.
- There is relatively little difference between the predictions of the cross-validated and non-verified forms of the two algorithms.
- Interestingly, there is little difference between the weighted and non-weighted k-NN methods (for $k = 10$).
- The results of the duplicate simulations are generally similar, with the largest difference being for the cross-validated bagged tree method.

Table 4-2. Comparison of active-passive classifications for 100 000 realizations using cross-verified and non-verified bagged tree and weighted k-NN algorithms.

	Bagged tree				Weighted k-NN (k = 10)				Unweighted k-NN (k = 10)
	Non-verified		Cross-verified		Non-verified		Cross-verified		Non-verified
	Simulation 1	Simulation 2	Simulation 1	Simulation 2	Simulation 1	Simulation 2	Simulation 1	Simulation 2	From Table 4-1
Time steps:									
% active	85	87	88	76	78	80	78	79	80
% passive	15	13	12	14	12	20	22	21	20
Realizations:									
% active only	84	86	87	73	43	43	43	43	57
% passive only	11	9	7	14	2	2	2	2	2

One advantage of the cross-validated versions of the algorithms is that a measure of the accuracy of the method can be obtained. The results from each of the n training sessions are averaged and the accuracy of the model determined. Using 5-fold cross-validation, the most accurate method was the bagged tree algorithm with an accuracy of 89.2 % (with a 0.60 % standard deviation), while the weighted k-NN algorithm had an accuracy of 77 %. Due to the random selection of the n -fold groups and implementation of the algorithm, the training of a data set varies slightly with each model run.

Another measure of algorithm performance is to plot the true versus the predicted class in the form of a so-called a confusion matrix. Figure 4-5 and Figure 4-6 show the confusion matrices for the weighted k-NN and bagged tree methods, respectively. In these plots, a green colour indicates that the predicted membership or class was the same as the actual (true) membership. A red colour indicates that the true and predicted classes were different. True-positives were higher for the bagged tree method, with false negatives half of the k-NN method. The different shades of green and red indicated level of accuracy for example a higher true positive rate is more green while a higher false negative rate is more red.

Because of its higher accuracy, the cross-validated bootstrap aggregation (bagged) decision tree classification method was used to determine active and passive conditions for the probabilistic pitting analysis in Section 5.

Within a given simulation (comprising up to one million realizations), the A-P classification is consistent for all realizations. In other words, a given set of environmental conditions will be classified as either A or P for all the realizations in which this data point is selected. However, the classification method is re-trained for each simulation, so that a given data point could be classified differently for replicate simulations.

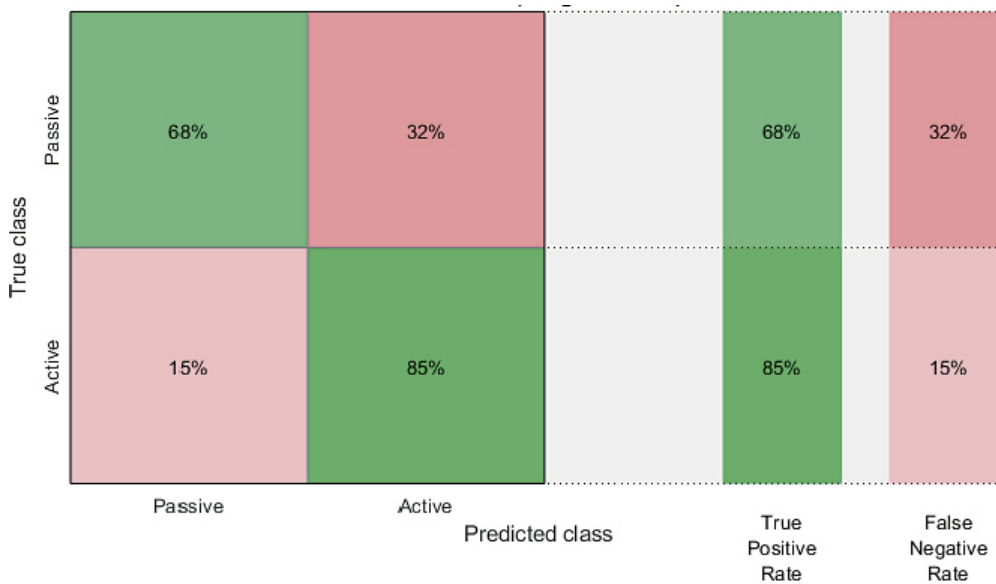


Figure 4-5. Confusion matrix comparing true class and predicted class for the weighted k-NN algorithm ($k = 10$). Class 0 is passive and class 1 is active. Overall accuracy was 77 % and colour shading is graduated to illustrate respective performance.

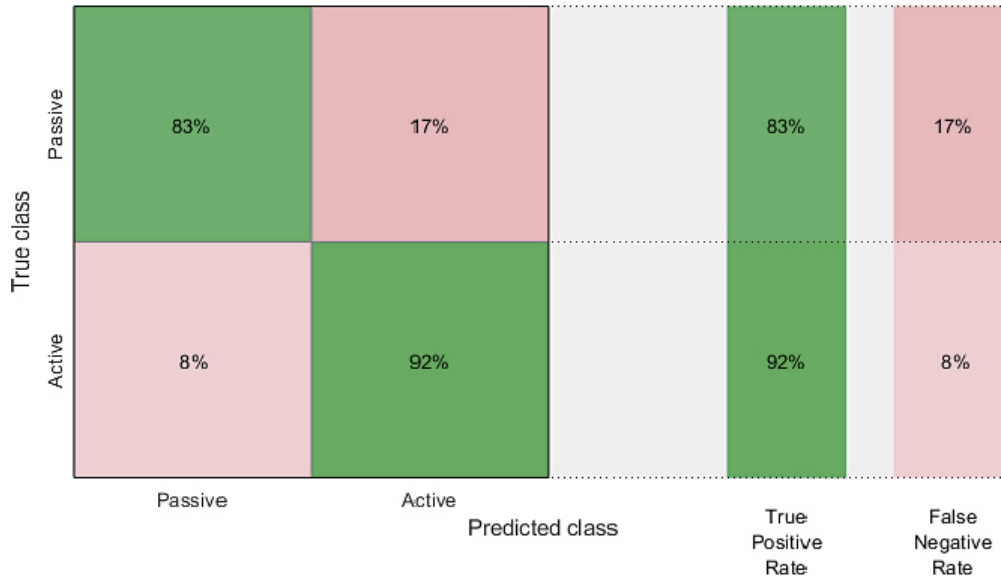


Figure 4-6. Confusion matrix comparing true class and predicted class for the bootstrap aggregation decision tree algorithm. Overall accuracy was 89.2 % and colour shading is graduated to illustrate respective performance.

4.3.2 Regression methods for critical potential data

As described in Section 3.1.2, there are two sources of critical potentials for systems exhibiting passive behaviour, those of Cong et al. (2009) and those from the work of Qin et al. (2017). Here the treatments of the two sets of data are described and the results of a comparison exercise are discussed.

Cong et al. (2009) used a linear regression method to fit their data to expressions involving pH, $[Cl^-]$, $[SO_4^{2-}]$, and $[HCO_3^-]$ (see Equations (3-1) and (3-2)). The effects of HCO_3^- and Cl^- were studied separately from those of HCO_3^- and SO_4^{2-} , leading to two expressions for E_B and E_{RP} . However, in the case of E_B , Cong et al. (2009) were able to combine the effects of all three anions into a single expression (Equation 3-1), but separate expressions were retained for the effects of SO_4^{2-} and Cl^- in the case of E_{RP} (Equations (3-2a) and (3-2b), respectively). The experimental studies were carried out at only two pH's (pH 8.3 and 9.5) and at a single temperature (room temperature, 25 °C).

For the probabilistic pitting analysis, the original expressions of Cong et al. (2009) were modified slightly. The combined expression for E_B was used in the original form, except that the first term was replaced by a variable denoted $E_{Bconstant}$ (Equation 4-1)

$$E_B = E_{Bconstant} + 0.116 \times \log[OH^-] + 0.197 \times \log[HCO_3^-] - 0.130 \times \log([SO_4^{2-}] + [Cl^-]) V_{SCE} \quad (4-1)$$

The mean value of the variable $E_{Bconstant}$ was set to 1.11 V_{SCE} as in the original expression of Cong et al. (2009). However, in the probabilistic analysis, $E_{Bconstant}$ was treated as a normally distributed parameter with a standard deviation of 0.061 V based on the distribution of replicate measurements shown for "E_B" in Figure 2-5. For each realization, a different value of $E_{Bconstant}$ was selected from the normal distribution in order to represent the uncertainty in the value of the breakdown potential. This source of variability reflects the variability in the replicate measurements in a single environment, and does not represent either the variability in the dependence of E_B on pH, $[Cl^-]$, $[HCO_3^-]$, and $[SO_4^{2-}]$, or the uncertainty due to the particular method used to determine the critical potentials.

A similar approach was used for the uncertainty in E_{RP} . In this case, however, because the effects of all four anions could not be combined into a single expression, the estimation of E_{RP} was based on the effects of pH, $[Cl^-]$, and $[HCO_3^-]$ only (Equation 4-2)

$$E_{RP} = E_{RPconstant} + 0.00373 \times \log[OH^-] - 0.0139 \times \log[HCO_3^-] - 0.0566 \times \log[Cl^-] V_{SCE} \quad (4-2)$$

Repassivation potentials calculated based on the alternative expression given by Cong et al. (2009) based on the pH, $[\text{SO}_4^{2-}]$, and $[\text{HCO}_3^-]$ produced similar E_{RP} values to the prediction based on Equation (4-2). As for the breakdown potential, the $E_{\text{RPconstant}}$ term was sampled for each realization using a normal distribution described by a mean of $-0.0925 \text{ V}_{\text{SCE}}$ (the original value of Cong et al. 2009) and a standard deviation of 0.005 V (corresponding to that for the E_{RP} data in Figure 2-5).

The second source of critical potential data is the study of Qin et al. (2017) comprising a set of approximately 200 individual E_{B} and E_{RP} data determined as a function of five environmental variables (temperature, pH, $[\text{Cl}^-]$, $[\text{HCO}_3^-]$, and $[\text{SO}_4^{2-}]$). A regression model was fitted separately to the E_{B} and E_{RP} data. As a starting point, the MathWorks MATLAB Regression Learner was used to test many possible regression methods, from which two specific regression methods were chosen.

The first method chosen was a machine learning algorithm called support vector machines (SVM) where learning algorithms are applied to numerical data sets for the purpose of regression and predicting outcomes (Vapnik 1995). Specifically, a Gaussian kernel function was used which returns a smooth regression function for prediction of breakdown and repassivation potentials. The second method chosen was a machine learning algorithm typically used in statistical analysis which uses a regression tree to predict real number outcomes. Specifically, a regression tree algorithm with least squares gradient boosting (Friedman 2001) was used where, at each step of the algorithm, a fit to minimize mean-square error is desired.

The method with the lowest root-mean-square-error (RMSE) was selected for use in the full probabilistic model. To illustrate the error in the chosen algorithms, Figure 4-7 and Figure 4-8 show the predicted vs. actual response for the E_{B} data from the Qin et al. (2017) dataset. A perfect algorithm would result in a 1:1 linear plot. The method selected for use in the model was the boosted tree method with a RMSE = 0.0278 (with a standard deviation of 0.0004) for E_{B} and a similar response for E_{RP} of RMSE = 0.0383 (with a standard deviation of 0.0009). As with the A/P classification methods, an n-fold cross-validation (with $n = 5$) was used for the implementation of regression methods. Thus, when using the Qin et al. critical potentials, there is no variation from realization-to-realization due to the regression method but, unlike the analysis for the Cong et al. dataset, there is also no variability due to the uncertainty in the data themselves. However, the regression fit of the Qin et al. data will vary from simulation to simulation.

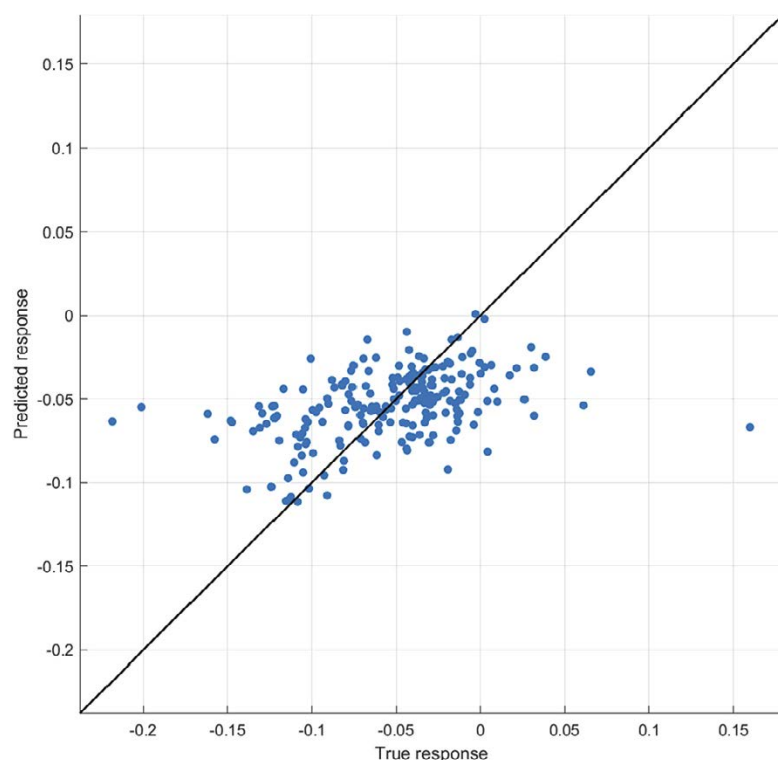


Figure 4-7. Predicted (y-axis) versus the measured (x-axis) breakdown potentials (in VSCE) for a Gaussian support vector machine regression algorithm of EB with RMSE = 0.0422.

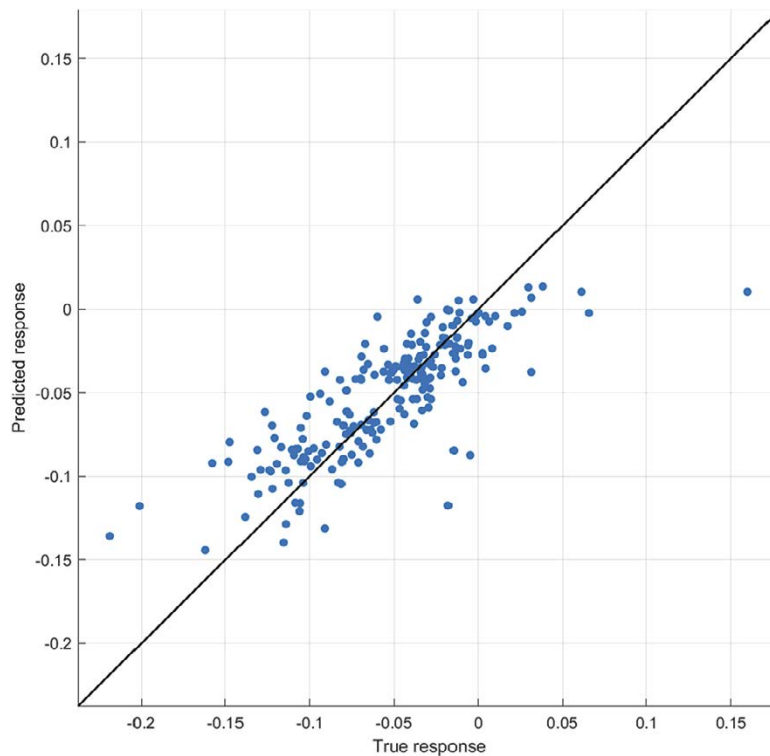


Figure 4-8. Predicted (y-axis) versus the measured (x-axis) breakdown potentials (in VSCE) for a boosted tree ensemble regression algorithm of EB with RMSE = 0.0278.

Preliminary simulations were performed to determine which critical potential dataset to use in the full probabilistic analysis. Table 4-3 shows a comparison of the two datasets based on the results of simulations comprising 100 000 realizations using the unweighted k-NN ($k = 10$) method for classification of A and P behaviour. As would be expected, there is no difference in terms of the number of time steps (first row) or the number of realizations (fourth row) exhibiting active or passive behaviour. Any slight differences are the result of differences in the selection of the temperature profile and pore-water pH and $[Cl^-]$ from the corresponding distributions, which are still apparent in a simulation comprising 100 000 realizations.

However, there are significant differences in the predictions using the two critical potential datasets in terms of pit initiation and growth. If the surface is passive, the Qin et al. (2017) data predict that initiation occurs 61 % of the time, whereas the Cong et al. (2009) data predict only 30 % initiation (second row, Table 4-3). Again, a slight difference might be expected as the E_{CORR} values will not have been exactly the same due to differences in T , $[Cl^-]$, and the value of the O_2 -consumption rate constant a , but such differences would be expected to be minor given the approximately 1.4 million time steps for which conditions were predicted to be passive. Of more interest, however, is the difference in the number of propagating pits. For the Cong et al. (2009) data, pit propagation only occurred for 0.2 % of the time steps following pit initiation (third row). In contrast, pits were predicted to propagate 100 % of the time following initiation and prior to repassivation based on the Qin et al. (2017) critical potential data. As a consequence, a total of 21 574 pits were predicted for the 100 000 realizations based on the Qin et al. data, but only 41 based on the Cong et al. critical potentials (fifth row, Table 4-3).

Table 4-3. Comparison of the two sets of critical potential data based on the results of preliminary probabilistic analyses.*

	Cong et al. (2009)		Qin et al. (2017)	
Time steps:				
active	5747 128	80.9 %	5707 977	80.3 %
passive	1352 872	19.1 %	1392 023	19.7 %
Timesteps:				
passive but $E_{CORR} < E_B$	948 734	70.1 %	539 451	38.8 %
passive and $E_{CORR} > E_B$	404 138	29.9 %	852 572	61.2 %
Time steps:				
$E_{CORR} > E_B$ and $E_{CORR} > E_{RP}$	960	0.2 %	852 572	100 %
$E_{CORR} > E_B$ and $E_{CORR} < E_{RP}$	403 178	99.8 %	0	0 %
Realizations:				
active only	56 468	56.5 %	56 941	56.9 %
passive only	1 710	1.7 %	1 875	1.9 %
active and passive	41 822	41.8 %	41 184	41.2 %
Total number of pits	41		21 574	
Mean time steps pit growth	22		39	

* Based on simulations with 100 000 realizations using the unweighted k-NN (k = 10) A/P classification method. Each realization ran to 100 years in 71 time steps, for a total of 7.1 million time steps per simulation. Both the number of time steps or realizations, and the percentage of the total are given for each category.

The cause of this difference in behaviour is inconsistencies between the values of E_B and E_{RP} predicted using the Cong et al. (2009) expressions. From an electrochemical standpoint, the value of E_B must be greater than E_{RP} because otherwise the voltammogram would not exhibit the behaviour characteristic of film breakdown and growth illustrated by the example in Figure 2-1. However, this was not the case for the Cong et al. data and in the majority of cases (99.8 % of the time according to the data in Table 4-3), the predicted E_{RP} was greater than E_B . This is illustrated in Figure 4-9 which shows the values of E_B and E_{RP} predicted by the Cong et al. data for a series of 100 realizations. Although the E_B and E_{RP} values for individual time steps are not apparent, it is clear from the distributions of the two parameters that in many cases $E_B < E_{RP}$. That this was not the case when using the Qin et al. (2017) critical potentials is shown in Figure 4-10.

There are a number of possible reasons for the inconsistent behaviour exhibited by the Cong et al. data. First, as the original authors themselves cautioned, the applicability of Equations (4-1) and (4-2) for environmental conditions outside of the ranges for which they were determined is uncertain. In particular, the repository environmental data cover a wider range of pH (pH 7–10) than used experimentally (pH 8.3 and 9.5 only). Second, values for $E_{Bconstant}$ and $E_{RPconstant}$ were selected from normal distributions, but no attempt was made to correlate the two parameters. Thus, in some realizations the selected $E_{Bconstant}$ value might have corresponded to the (mean – 2s), for example, while the $E_{RPconstant}$ value might have corresponded to the (mean + 2s). Even in such an extreme eventuality, however, the value of $E_{Bconstant}$ ($0.988 V_{SCE} = 1.11 V_{SCE} - 2 \times 0.061 V$) would still have been more than 1 V more-positive than $E_{RPconstant}$ ($-0.0825 V_{SCE} = -0.0925 V_{SCE} + 2 \times 0.005 V$). Lastly, it is possible that taking into account the effect of $HCO_3^- - Cl^-$ on E_{RP} while ignoring the effect of $HCO_3^- - SO_4^{2-}$ results in an error in the predicted repassivation potential. While some discrepancy is to be expected, it should be pointed out that King and Lilja (2013, 2014) encountered the same issue when using the Cong et al. data regardless of whether they used Cong's Equation (3-2a) (for the $HCO_3^- - SO_4^{2-}$ system) or (3-2b) (for the $HCO_3^- - Cl^-$ system) to estimate the repassivation potential.

Regardless of the cause of the inconsistency when using the Cong et al. (2009) fitted expressions, it is apparent that the Qin et al. (2017) critical potential data provide a consistent dataset and it is this source of critical potentials that is used for the full probabilistic analysis.

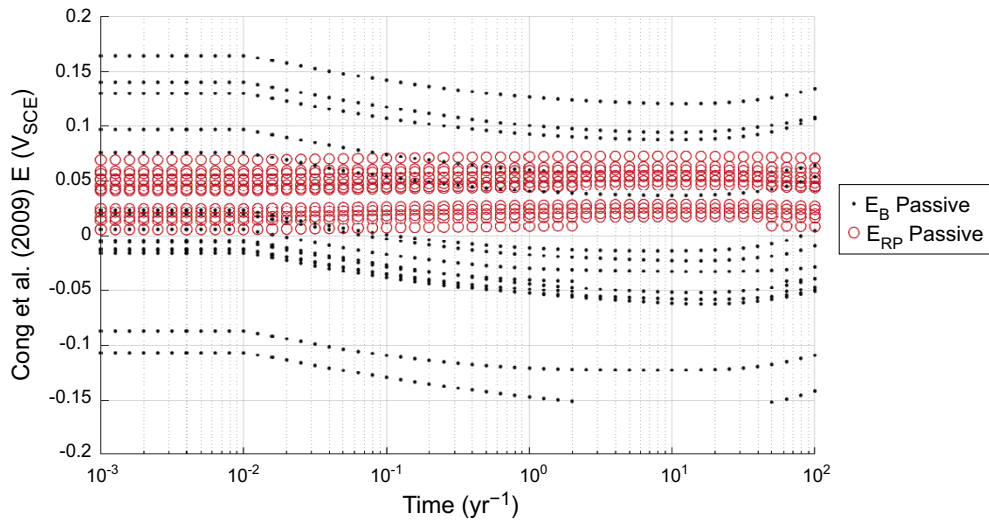


Figure 4-9. E_B and E_{RP} for a simulation with 100 realizations over 100 years using Cong et al. (2009) equations. The break in some of the curves correspond to times when, for the those particular realizations, the surface transitioned from passive to active and back to passive again. The critical potentials are not defined for active conditions.

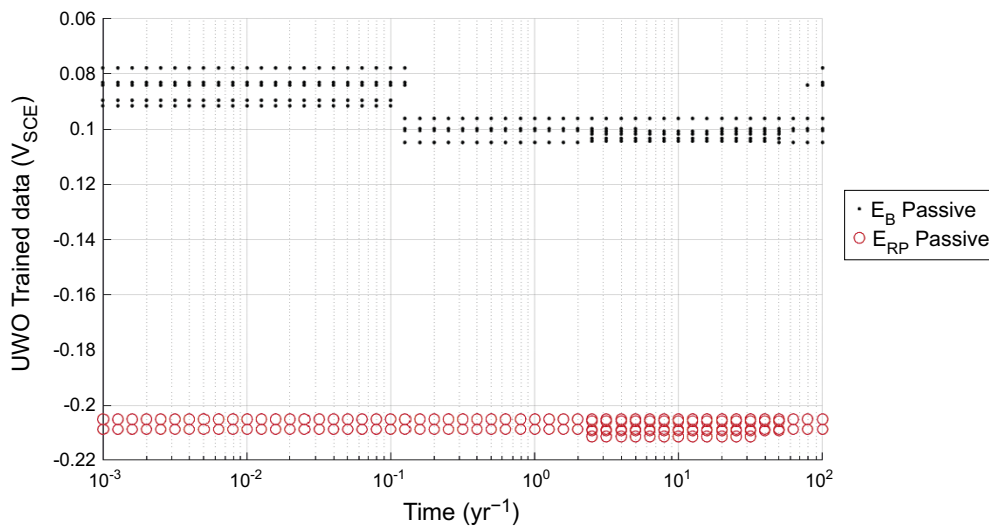


Figure 4-10. E_B and E_{RP} for a simulation with 100 realizations over 100 years using Qin et al. (2017) data.

4.3.3 Pit growth kinetics

In fitting Equation (3-3) to the long-term exposure data, Denison and Romanoff (1950) implicitly assumed that pit growth was continuous during the exposure. Five different periods of exposure were reported between 2 and 14.3 years. However, it is not clear *a priori* whether the evolution of environmental conditions in the repository will result in a single or multiple pitting events. For instance, passivation might be more likely at lower temperatures (due in part to the retrograde solubility of CaCO_3) so it is conceivable based on the temperature profiles in Figure 3-2 that the surface could be passive initially, then become active as the temperature increases, before passivating again as the canister cools. Whether such passive-active-passive transitions would result in multiple pitting events would also depend, of course, on the value of E_{CORR} which will tend to decrease with time as O_2 is consumed. Nevertheless, it is not impossible that there could be realizations in which there were multiple pitting events. In such circumstances, it would then be necessary to decide whether previous pits reactivate or whether new pits initiate at each pitting event. Furthermore, in the case of reactivated pits, it would be necessary to define whether growth re-starts at the initial rate or at the rate that existed at the end of the previous pitting event.

The possibility of multiple pitting events (i.e., re-initiation of a pit on a surface that has previously repassivated) in a given realization was investigated in the preliminary runs using the k-NN A/P classification method. A series of five replicate simulations (each comprising 100 000 realizations) was performed for each of the three pit-growth kinetics expressions described in Section 3.1.3 (the pitting factor approach, and the two time-dependent expressions for different soils from the Denison and Romanoff study). These fifteen simulations were performed for each source of critical potentials (the Cong et al. and Qin et al. datasets) for each of the k values previously considered (see Table 4-1).

The results of the analysis are summarized in Table 4-4. When using the Cong et al. critical potential data, at least one of the 100 000 realizations exhibited multiple pitting events in the majority of simulations. The fraction of simulations with at least one multiple pitting event tended to decrease with increasing k-value, which presumably is related to the tendency towards more active behaviour with increasing k-value characteristic of this classification method (see Table 4-1). In contrast, none of the simulations in which the Qin et al. critical potential data were used exhibited a single realization with multiple pitting events, regardless of the k-value. Thus, a second initiation event was not predicted in any of the total of 10.5 million realizations (fifteen 100 000-realization simulations for each of the seven k-values) using the Qin et al. data. Since this dataset of critical potentials has been selected over those of Cong et al. for the full probabilistic analysis (see Section 4.3.2), the possibility of multiple pitting events in a single realization is considered to be minimal and no further consideration was given to the issue of pit-reactivation versus initiation of a new pit.

Table 4-4. Fraction of the fifteen replicate simulations in which at least one realization exhibited multiple pitting events as a function of k-value for the two different sources of critical potential data.*

	k = 1	k = 2	k = 4	k = 6	k = 8	k = 10	k = 12
Cong et al.	14/15	15/15	15/15	10/15	8/15	5/15	2/15
Qin et al.	0/15	0/15	0/15	0/15	0/15	0/15	0/15

* Each of the simulations comprised 100 000 realizations with A/P classification using the unweighted k-NN method. The SVM regression method was used to estimate E_b for the Qin et al. data

Preliminary analyses were also conducted to determine the effect of the three different pit-growth expressions on the predicted damage. Five replicate simulations (each comprising 100 000 realizations) were carried out for each of the three pit-growth expressions using the Qin et al. critical potential data. The unweighted k-NN classification method was used to determine active or passive behaviour, with k varied from 1 to 12, as above. The three pit-growth rate expressions were:

- Constant pitting rate of 5 $\mu\text{m}/\text{yr}$ based on a pitting factor of 5 and a uniform corrosion rate of 1 $\mu\text{m}/\text{yr}$.
- Soil #65 of the Denison and Romanoff study ($A = 89.2 \pm 46 \mu\text{m yr}^{-n}$, $n = 0.57 \pm 0.26$).
- Soil #66 of the Denison and Romanoff study ($A = 188 \pm 41 \mu\text{m yr}^{-n}$, $n = 0.12 \pm 0.11$).

For the two-time-dependent pit-growth expressions, both A and n (Equation (3-3)) were assumed to be normally distributed with the \pm indicating one standard deviation. Thus, whereas the pit growth rate was the same for all realizations in the case of the “constant” expression, the rate would vary from realization-to-realization for the two Denison and Romanoff expressions. These two soils were selected partly because of their different time-dependences. For pitting periods of less than 5.2 years, the expression for soil #66 results in greater pit depths, while the expression for soil #65 predicts deeper pit at longer times.

The maximum predicted pit depths for the three pit-growth expressions are compared in Table 4-5. These depths represent the maximum pit depth in a simulation and are taken from the same series of 100 000-realization simulations as in Table 4-4. Five replicate simulations were performed for each growth expression for each k-value. The two time-dependent pit-growth expressions from the Denison and Romanoff study predict the largest maximum pit depths, with the expression for soil #65 giving the highest values. This implies that the maximum pitting period is greater than 5.2 years, which is the time at which the two expressions would give the same time-averaged rate. The \pm in the table represents the standard deviation of the five replicate simulations. In contrast, the maximum pit depth for the constant

pit-growth rate expression was predicted to be 32 μm for all simulations, representing a maximum period of pit growth of 6.3 years (based on a constant pitting rate of 5 $\mu\text{m}/\text{yr}$). A period of 6.3 years (or $10^{0.8}$ years) is equivalent to the initial 48 time steps (see Section 4.3.4).

Table 4-5. Predicted maximum pit depth (in μm) at the time of repassivation for the three pit-growth expressions for the same simulations summarized in Table 4-4.

	k = 1	k = 2	k = 4	k = 6	k = 8	k = 10	k = 12
Constant	32	32	32	32	32	32	32
Soil #65	884 \pm 69	970 \pm 120	1010 \pm 220	972 \pm 188	940 \pm 143	941 \pm 216	843 \pm 168
Soil #66	689 \pm 32	698 \pm 38	695 \pm 29	686 \pm 28	679 \pm 14	693 \pm 45	649 \pm 9

The values in Table 4-5 are the average of the maximum pit depths for each simulation. The pit depth also varies within a given simulation, however, because of different pitting periods for each of the 100 000 realizations and, for the two soils expressions, because of differences in the pitting kinetics. The distribution of predicted pit depths within a given simulation is discussed in more detail in the Section 5.

Since the pit-growth expression for soil #65 produces the largest maximum pit depths, this expression was used for the full probabilistic analysis described in Section 5.

4.3.4 Time discretisation

The time range used in the model runs from 10^{-5} years through to 100 years. The lower bound (corresponding to a period of approximately 5 minutes) is considered to represent the time of emplacement, with the upper bound considered to exceed the maximum time at which all of the initially trapped O_2 has been consumed. In numerical analysis, a sufficiently small time step is required to ensure accuracy, but not so small that the execution time becomes excessive. Stepping through the defined time range in linear increments would be problematic due to the 7 orders of magnitude (also referred to as 7 generations). Instead, we step through with equal increment over each generation, in other words in equal increments of the logarithm of time ($\log_{10} t$). For example the range of each generation could be split into 10 even increments and there would be 10 increments between 10^{-5} and 10^{-4} years and similarly 10 increments between 10 and 100 years. The result is the absolute numerical time increment is changing over time.

The discretization of the time scale has been chosen to be sufficiently small such that further numerical time discretization does not affect the solution. The results from the selected discretizations is then said to be a solution independent of the time. To determine the minimum time step, the results of a simulation are analysed until no changes occur with decreasing time step. Pit depth distribution statistics were chosen as the metric to compare simulations with varying time discretization. The number of time steps per generation (per each decade of time) was varied between one and 1 000, and the effect on the pit depth distributions determined for simulations comprising 100 000 realizations.

The pit-depth distributions for the different time-step discretizations are shown in Figure 4-11, with the mean and standard deviations given in Table 4-6. Based on the values of the mean pit depth, the “fine” discretization (100 time steps per decade) differs from the “finer” discretization (1 000 time steps per decade) by only 0.8 %. The difference between the fine and “normal” (10 time steps per decade) is 5.2 %, with the mean pit depth predicted using a “coarse” (1 time step per decade) discretization clearly being too coarse.

Table 4-6. Pit distribution statistics for various time discretizations (based on 100 000 realizations).

Discretization	Coarse	Normal	Fine	Finer
Time steps per generation	1	10	100	1000
Mean pit depth (m)	1.15×10^{-4}	5.71×10^{-5}	5.39×10^{-5}	5.43×10^{-5}
Pit depth standard deviation (m)	1.46×10^{-4}	5.77×10^{-5}	5.41×10^{-5}	5.31×10^{-5}

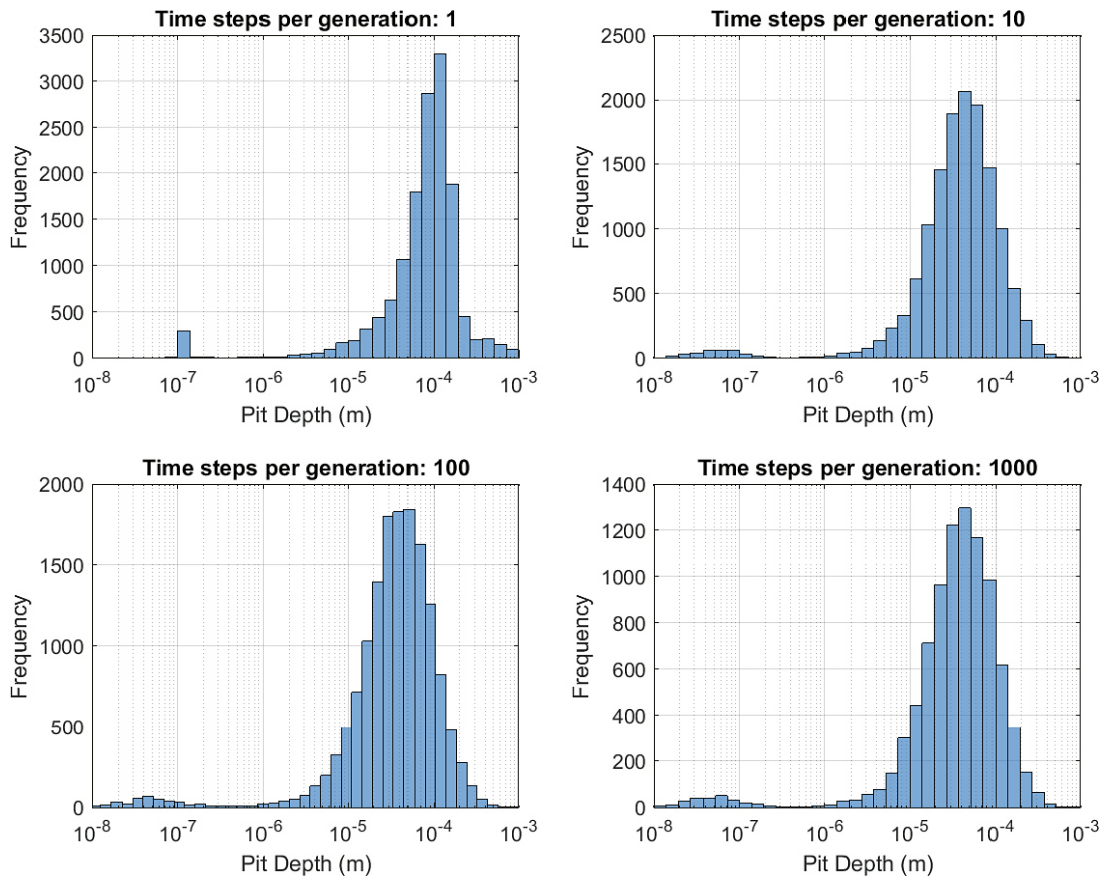


Figure 4-11. Pit depth histograms for varying log time step discretizations.

Based on these results, the normal discretization (10 time steps per decade) was used for the purposes of code development, but the fine discretization (100 time steps per decade) was used for the full probabilistic analysis. A 1 000 000 realization Monte Carlo simulation using the fine discretization takes 10 days to run using 20 processors and uses approximately 100 GB of memory. Running the same model at the normal discretization takes only 1 day. Repeatability of the results can be observed by comparing multiple runs of models with the same input parameters. The standard deviation of the mean pit depth over five independent model runs was 1.15×10^{-6} m with a mean of 5.39×10^{-5} m.

4.4 Final version of the model as implemented

4.4.1 Sources and treatment of input data

As described above, a number of different sources of input data and of methods of analysis have been considered during the development of the model. Based on these preliminary studies, the full probabilistic analysis described in Section 5 is based on the following sources of data and statistical analytical methods:

- Classification of active or passive behaviour is based on the verified bootstrap aggregating decision tree (“bagging”) method.
- Regression fitting of the critical potential data of Qin et al. is performed using the boosted tree ensemble method.
- Pit-growth kinetics are based on the Denison and Romanoff expression for soil #65.
- “Fine” time discretization is used with 100 time steps per decade, from 10^{-4} to 100 years.

All other input data are as described in Section 3.

4.4.2 Simulations performed

A number of types of simulation were performed in order to illustrate various aspects of the predicted pitting behaviour:

- A “full probabilistic analysis” was run using the data and statistical methods described above with a total of 1 000 000 realizations per simulation. The full analysis, therefore, is equivalent to an average of approximately 140 realizations for each of the approximately 6 000 deposition holes in the repository.
- In order to present the model output visually, a minimal probabilistic analysis was performed with a total of just 100 realizations with a complete set of diagnostic graphical plots. For simulations with a larger number of realizations, a limited set of graphical outputs were produced in order to avoid the appearance of an overwhelming mass of data points.
- The progression of the pit-depth distribution with time was investigated using 100 000-realization simulations for periods of between 10^{-4} and 0.1 years in order to examine the development of shallow pits.
- The full analysis was performed with the maximum value of the triangular pH distributions of either pH 10 or pH 9.
- Duplicate simulations were performed with 1 000 000 realizations and five simulations were performed with 100 000 realizations in order to examine the reproducibility of the simulations.

5 Results of probabilistic pitting analysis

5.1 Full probabilistic analysis

The full probabilistic analysis comprised a simulation with 1 000 000 Monte Carlo realizations. A summary plot of the evolution of active and passive behaviour is shown in Figure 5-1 and a histogram of the predicted pit depths in Figure 5-2. Table 5-1 provides a summary of various indicators of the active-passive and pitting behaviour, not only for the full probabilistic analysis comprising 1 000 000 realizations but also for more limited simulations with either 100 000 or 10 000 realizations.

Figure 5-1 shows the time dependence of the number of realizations exhibiting active behaviour (in blue), passive behaviour without film breakdown (in black), and passive behaviour with pit growth (in red). At any given time, between 85 % and 90 % of the realizations predict that the surface of the canister will be in the active condition based on the sampled pH and $[\text{Cl}^-]$ and the time-dependent T , $[\text{SO}_4^{2-}]$, and $[\text{HCO}_3^-]$. This prediction is consistent with our general understanding that copper will corrode actively in the presence of compacted bentonite (King et al. 2010). For the remaining 10–15 % of the realizations, the canister surface is predicted to be passive. Film breakdown leading to pitting occurs early during the evolution of the repository environment, at a time when there is sufficient O_2 present to shift E_{CORR} to a value more-positive than E_{B} . Pitting becomes infrequent at times beyond 2–3 years, although it is not possible to state when pitting finally ceases based on the scale used in Figure 5-1. After 2–3 years, passive conditions become marginally more favoured, possibly because of the decrease in $[\text{SO}_4^{2-}]$ with the increasing canister temperature during this period. Sulphate exhibits a complex effect on the active-passive behaviour of copper (Qin et al. 2017). This tendency towards more-passive conditions is not a consequence of an increase in $[\text{HCO}_3^-]$ or decrease in temperature, both of which promote passivation (Qin et al. 2017), since both parameters are evolving in the opposite direction during this pre-thermal-peak period.

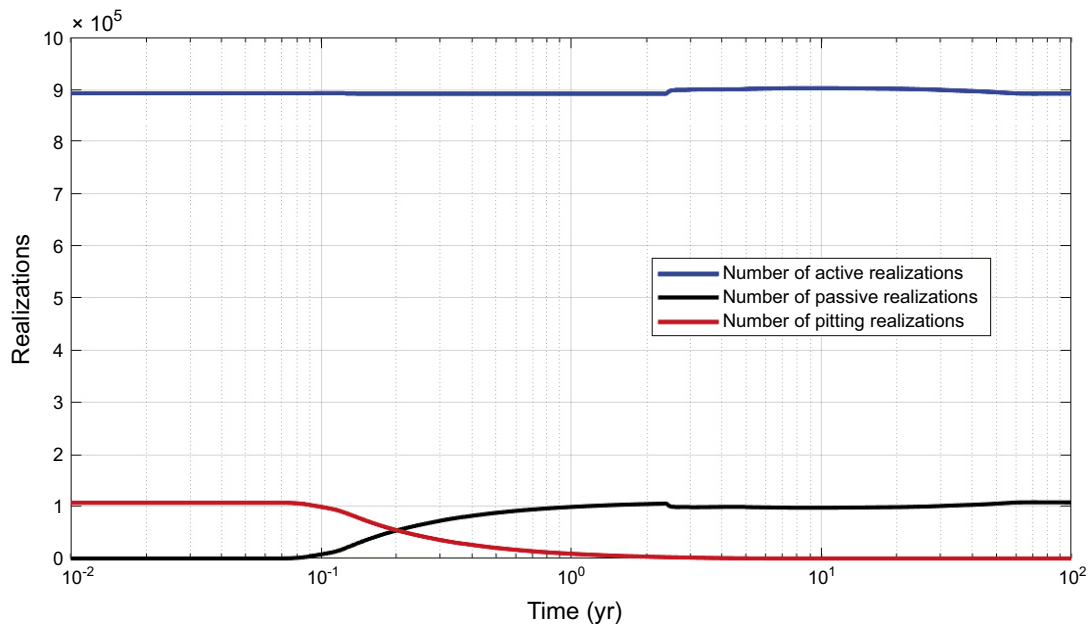


Figure 5-1. Time dependence of the number of active, passive, and pitting realizations for the full probabilistic analysis.

The consequences for pitting of the canister are shown in Figure 5-2 in the form of a histogram of predicted pit depths. The maximum pit depth for any of the 1 000 000 realizations is of the order of 1 mm, with the majority of pits being < 100 μm deep. (An allowance of the order of 100 μm was used to account for localised corrosion in the form of surface roughening for SR-Site, SKB 2010). On the logarithmic depth scale used in Figure 5-3, the pit-depth histogram exhibits a bi-modal distribution, the cause of which is discussed in more detail in Section 5.3.

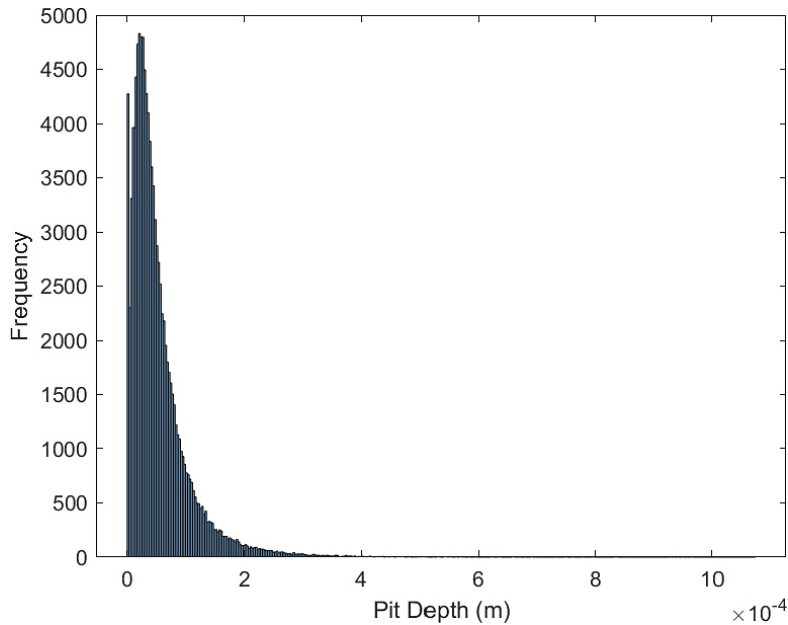


Figure 5-2. Histogram of pit depths for the full probabilistic analysis. The majority of pits are predicted to be < 100 μm (1×10^{-4} m) in depth.

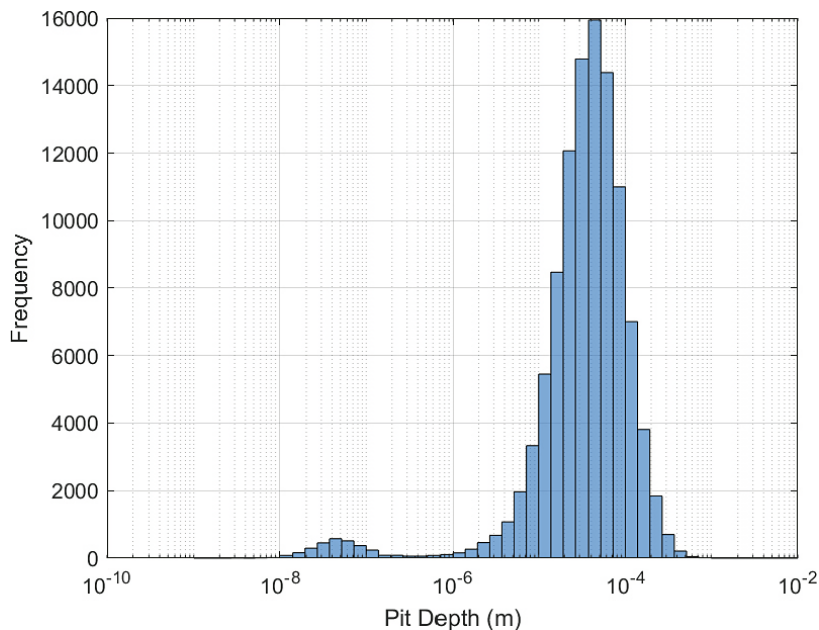


Figure 5-3. Semi-log histogram of pit depths for the full probabilistic analysis.

The duration of the individual pitting events also varied significantly (Figure 5-4). The shortest pitting period was 0.0661 years (24 days) and the longest period was 6.92 years. However, there were relatively few long-lived pits, with the median and mean pit lifetimes being 0.20 and 0.41 years, respectively.

Analysis of the pit initiation times is also interesting. Of the total of 124289 pits, all but 104 of them (124185 or 99.92 % of the total) initiated immediately (i.e., at the minimum time of 10^{-5} years). Of the pits that initiated later, 2 pits initiated at 0.126 years and the remaining 102 pits initiated between 2.5 and 3.0 years. As discussed in more detail below, this initiation behaviour is consistent with the observation that active or passive behaviour tends to be established for the entire simulation period, with relatively few instances of switching between active and passive behaviour, or vice versa, during a given realization.

Greater insight into the predicted behaviour can be obtained from some of the statistical data presented in Table 5-1. Data are presented for both individual time steps (of which there are 701 for each realization) and on the basis of the number or percentage of realizations. The canister surface is predicted to be passive for only 12.8 % of the 701 million time steps in the full probabilistic analysis. Interestingly, active conditions were maintained throughout the 100-year simulation period for 83.5 % of the realizations. In contrast, 10.9 % of the realizations were entirely passive, with only 5.6 % of the realizations exhibiting both active and passive conditions.

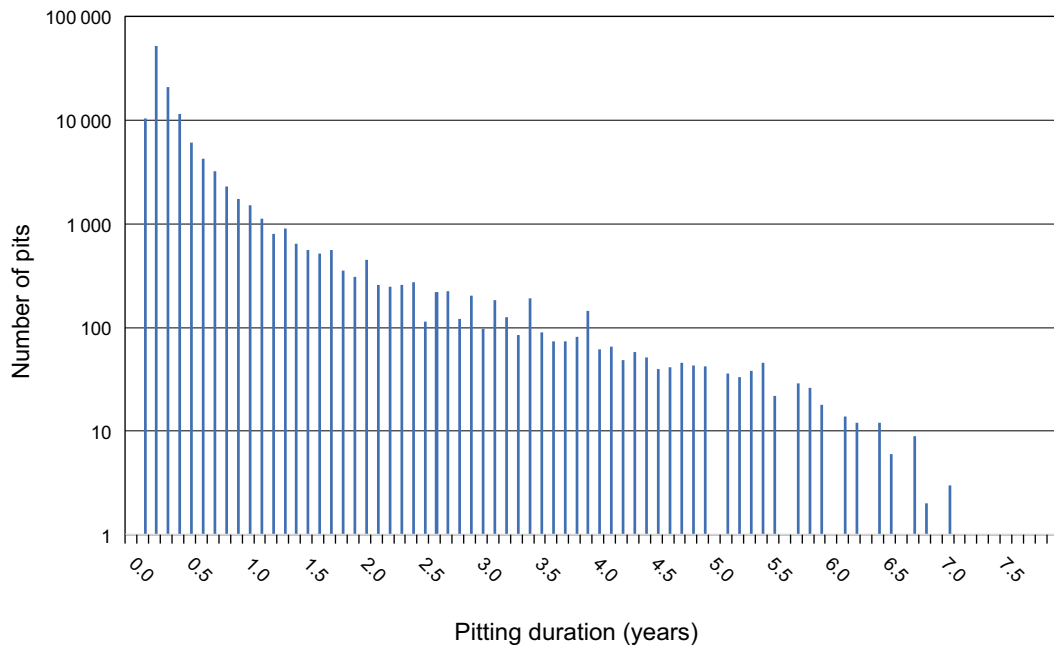


Figure 5-4. Distribution of pitting durations for the full probabilistic analysis.

Table 5-1. Summary statistics for the full probabilistic pitting analysis and comparison with simulations with smaller numbers of realizations.*

Parameter	Full probabilistic analysis		
	1 000 000	100 000	10 000
% of time steps:			
Active condition	87.2	87.2	87.1
Passive condition	12.8	12.8	12.9
% of realizations:			
Active only	83.5	86.1	81.4
Passive only	10.9	9.8	10.4
Active and passive	5.6	4.1	8.3
% of passive time steps:			
No film breakdown ($E_{CORR} < E_B$)	39.0	35.6	44.9
Pit initiation and growth	61.0	64.4	55.1
Film breakdown but no growth ($E_{CORR} > E_B$, but $E_{CORR} < E_{RP}$)	0.0	0.0	0.0
% of all realizations in which:			
Pit initiation occurred	12.4	13.3	13.7
Pit re-initiation occurred	0.0	0.0	0.0
Predicted pit depth < 100 μm	10.8	11.6	11.5
Predicted pit depth > 1 mm	0.0002	0.00010	0.0000
Total number of pits	124 289	13 340	1 372
Number of pits with depth < 100 μm	108 415	11 575	1 154
Number of pits with depth > 1 mm	2	1	0
Maximum pit depth (mm)	1.22	1.19	0.61
Median pit depth (mm)	0.039	0.039	0.043
Mean pit depth (mm)	0.053	0.054	0.060
Pit depth standard deviation (mm)	0.053	0.054	0.059

* All simulations were performed for a period of 100 years with 100 time increments per decade (total of 701 time increments). The critical potential data of Qin et al. were used, along with the pit-growth kinetics for soil #65.

Of the 12.8 % of the time that the surface was predicted to be passive, no film breakdown occurred 39.0 % of the time (third row of data in Table 5-1). The passive film was predicted to remain stable in these cases either because the film-breakdown potential was quite positive for the environmental conditions at the time or because the value of E_{CORR} was too low possibly because a fraction of the O_2 had already been consumed. For the remaining 61.0 % of the time that the surface was passive, however, pit initiation and propagation were predicted to occur. For none of the approximately 55 million individual time steps that the surface was passive was film breakdown predicted but no growth occurred, i.e., in no case was $E_{CORR} > E_B$ and $E_{CORR} < E_{RP}$. In other words, in all cases $E_B > E_{RP}$ as would be expected based on the definition of these parameters and which was confirmed during preliminary simulations when using the Qin et al. critical potential data, but not those of Cong et al. (see the earlier discussion in Section 4.3.2).

It is also of interest to consider the predicted pitting behaviour on the basis of the entire realizations, each of which represents the evolution of the corrosion behaviour of a canister in a particular deposition hole (fourth row of data in Table 5-1). Pitting is predicted to occur in 12.4 % of the cases, but in no case is more than one pit predicted to initiate in a given realization. Therefore, the evolution of environmental conditions does not support the repeated passivation and breakdown of a surface film on the canister. Once conditions for repassivation are established, there is no indication that the environment can once again support initiation and growth of a pit.

In terms of pit growth (fifth row of data in Table 5-1), the vast majority of pits (87 %) are predicted to be < 100 μm in depth. As noted above, this depth is of the same order as the corrosion allowance for surface roughening used for SR-Site (SKB 2010). Only two of the total of over 124 000 predicted pitting events resulted in pit depths > 1 mm. The deepest pit was predicted to be 1.22 mm deep, with a median and mean depth of only 39 μm and 53 μm , respectively. As indicated in Figure 5-3, there is a relatively wide distribution of pit depths, consistent with the standard deviation of 53 μm .

Also given in Table 5-1 are summary statistics for simulations with a smaller number of realizations than the full analysis, in particular 100 000 and 10 000 realizations. Although there are slight differences between the results of the three simulations, the overall conclusions are similar for each. The most significant difference is the increase in maximum pit depth and a higher probability of predicting a deep pit (> 1 mm) with increasing number of realizations.

5.2 Visualization of pitting behaviour based on 100 realizations

It is difficult to get a clear picture of the underlying causes of the predicted active-passive behaviour and resulting pitting from a simulation consisting of one million realizations. For this reason, a simulation was performed with just 100 realizations and the results illustrated graphically by comparing the responses of several parameters simultaneously. Figure 5-5 shows that the time dependence of active, passive, and pitting behaviour for a reduced number of realizations is similar to that for the full probabilistic analysis in Figure 5-1, suggesting that conclusions drawn from the 100-realization simulation should also be valid for the full analysis.

Figure 5-6 illustrates the interaction matrix for the five environmental parameters and time and shows their impacts on the active (blue) and passive (red) behaviour for a 100-realization simulation. If at least one of the two parameters varies with time (such as temperature, $[\text{SO}_4^{2-}]$, or $[\text{HCO}_3^-]$), the interaction diagram takes the form of 100 lines of red or blue (or, in a small number of cases, red and blue) symbols. If one of the two parameters is pH or $[\text{Cl}^-]$, both of which are constant for a given realization, the lines are straight, otherwise they are curves. The interaction diagram for pH and $[\text{Cl}^-]$, both of which are constant with time, is simply a plot of 100 red or blue symbols.

Such a large amount of data can be overwhelming, but a few trends are visible. The two environmental parameters that have the greatest influence on whether the surface is active or passive are the pH and $[\text{Cl}^-]$. Thus, the pH- $[\text{Cl}^-]$ and $[\text{Cl}^-]$ -pH interaction diagrams both show that passivation occurs at high pH (greater than \sim pH 8) and at $[\text{Cl}^-]$ less than approximately 0.05 mol/L. Active conditions can exist at pH > 9 if the $[\text{Cl}^-]$ is sufficiently high. In contrast, both active and passive behaviour is predicted over the entire ranges of T, $[\text{SO}_4^{2-}]$, and $[\text{HCO}_3^-]$ implying that these parameters exert either a secondary or negligible influence on passivity.

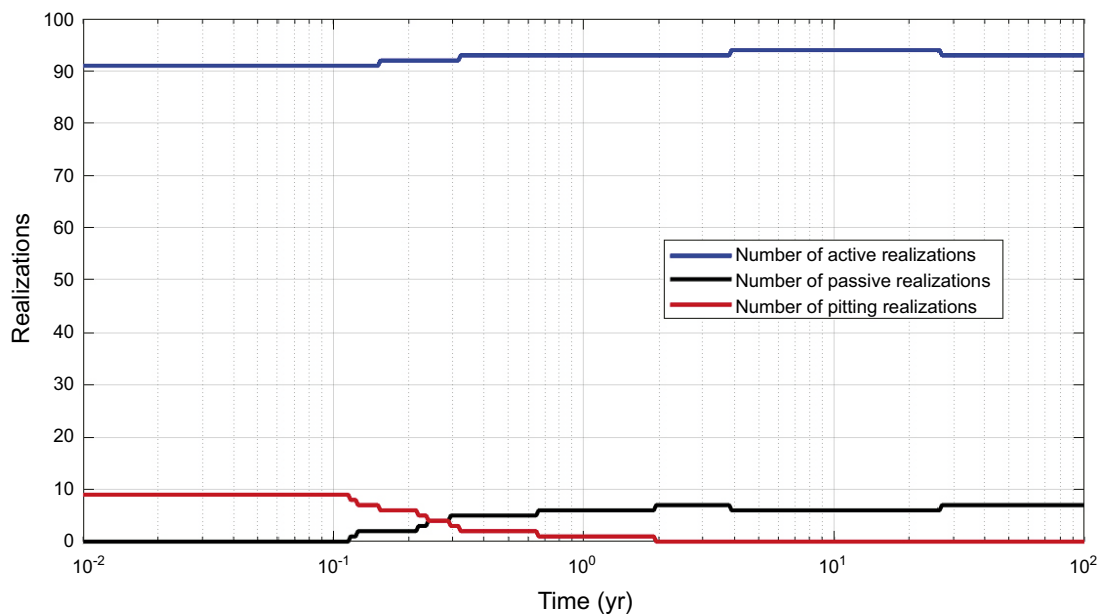


Figure 5-5. Time dependence of the number of active, passive, and pitting realizations for a simulation comprising one hundred realizations.

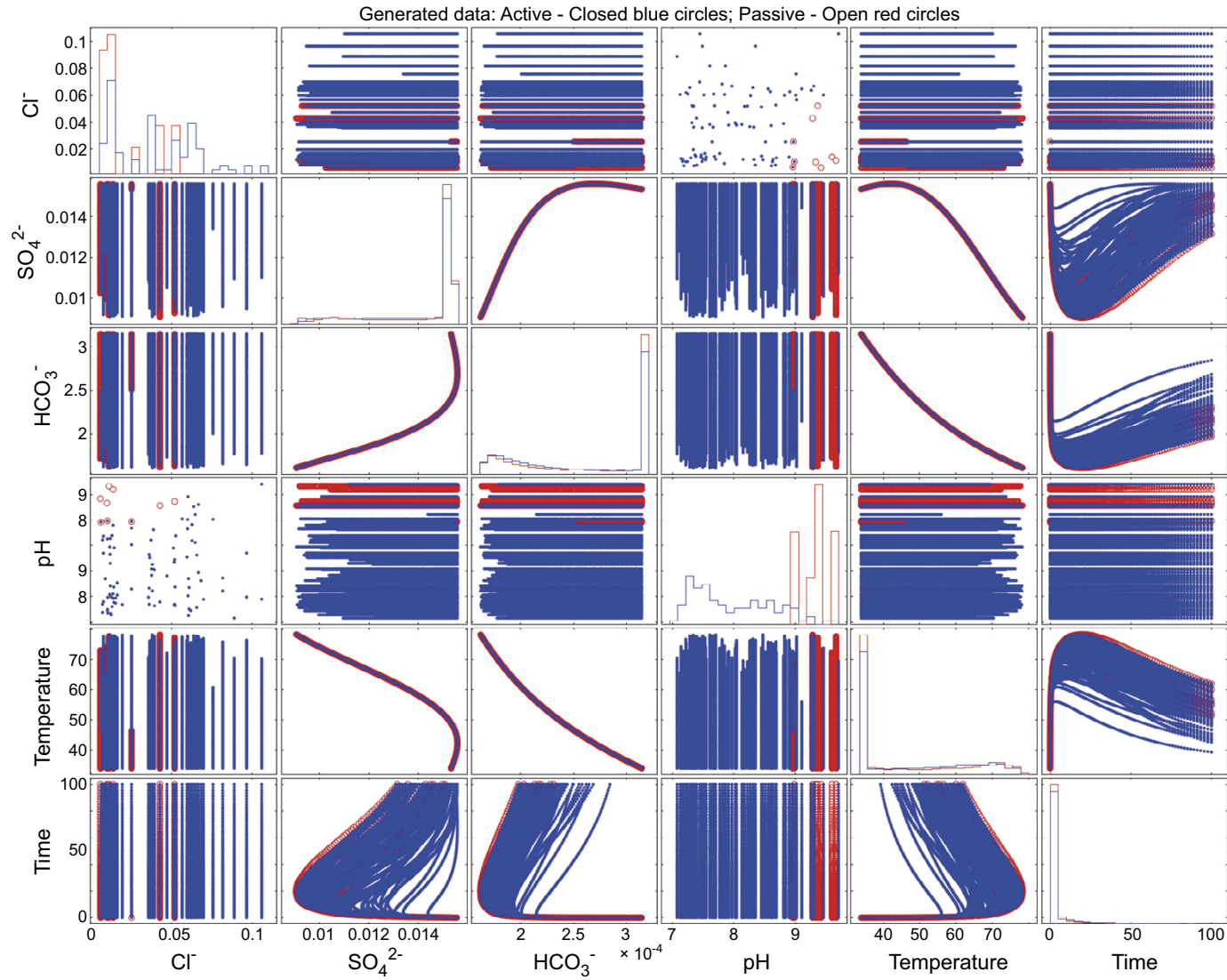


Figure 5-6. Interaction matrix showing the effects of different environmental parameters and time in determining either active (in blue) or passive (in red) behaviour.

Figure 5-7 shows the effect of pH, $[\text{Cl}^-]$, and temperature on the active-passive behaviour. The solid lines of points are the modelled conditions and the open circles are the experimental data of Qin et al. (2017), with blue and red indicating active and passive conditions, respectively. As shown by the interaction matrix, passivity tends to occur at higher pH and lower $[\text{Cl}^-]$, but tends to be independent of temperature. The evolution of the system with time is illustrated in Figure 5-8, which shows that the majority of realizations exhibit either active or passive behaviour. Although it is again difficult to visualise because of the large amount of data, there only appears to be one out of the 100 realizations in which both active and passive behaviour are predicted to occur. For this realization, labelled “a” in the figure and located at \sim pH 8.9 but at a low $[\text{Cl}^-]$, the canister surface is predicted to start off passive, then become active during the thermal peak, before finally passivating again. It is uncertain whether this switch to active behaviour is a direct result of the increase in temperature or because of the associated decrease in $[\text{HCO}_3^-]$ (or $[\text{SO}_4^{2-}]$). But the tenuous nature of the boundary between active and passive behaviour under these environmental conditions is indicated by the second realization labelled “b” in Figure 5-8. This realization is at the same pH but a slightly lower $[\text{Cl}^-]$, which should promote greater passivity, whereas in fact the canister surface is predicted to be active for the entire duration. It is quite possible that for this run the canister was generally hotter than for realization “a”, with a corresponding lower $[\text{HCO}_3^-]$ and a greater tendency to remain active.

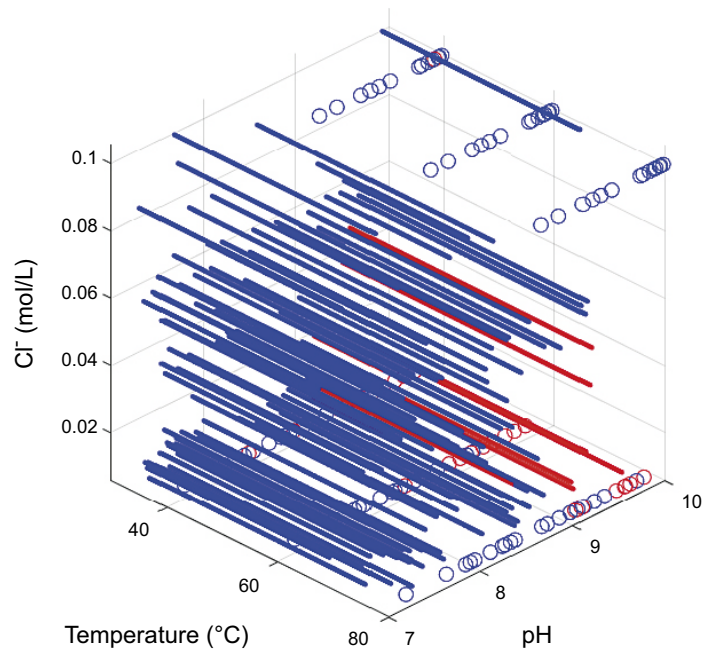


Figure 5-7. Effect of chloride concentration, pH, and temperature on the active (blue)-passive (red) behaviour. Lines of solid points represent the sampled data from the 100 realizations, with the corresponding experimental data of Qin et al. shown as open circles using the same colour coding.

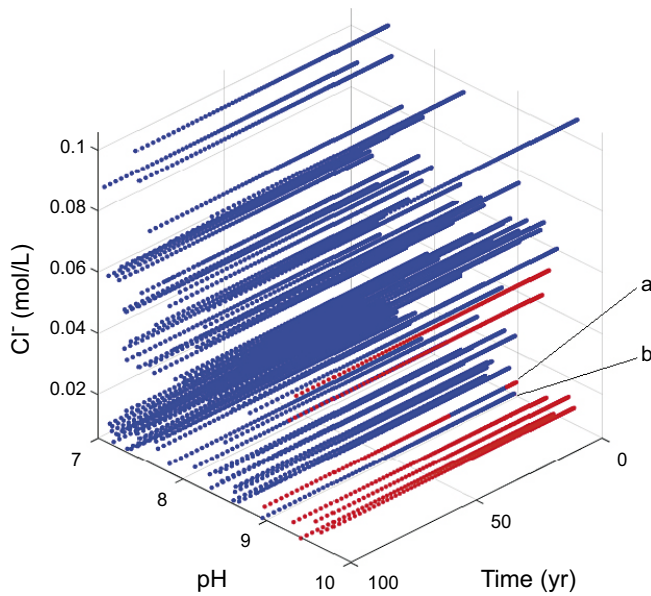


Figure 5-8. Effect of chloride concentration and pH on the time dependence of active (blue)-passive (red) behaviour. The lines of solid points represent the sampled data from the 100 realizations. Lines labelled a and b are discussed in the text.

Figure 5-9 shows the influence of both temperature and pH on passivity and confirms the over-riding effect of the latter environmental parameter. Passivation is only predicted for $\text{pH} \geq 8.9$ but, as noted above, active behaviour can be maintained at pH values as high as approximately pH 9.7, presumably because of a high $[\text{Cl}^-]$ (compare with Figure 5-7 and Figure 5-8). The experimental data indicating passivation at $\text{pH} < 8.9$ (as indicated by the red open circles in the figure) were determined at low $[\text{Cl}^-]$ or in the absence of chloride ions; conditions that are not representative of the predicted conditions in the repository. There is no indication from the data in Figure 5-9 that temperature has any influence on the active-passive behaviour. This conclusion is further shown in Figure 5-10 which shows evidence of passivity at all temperatures between 33 °C and the peak temperature for any of the profiles of approximately 79 °C.

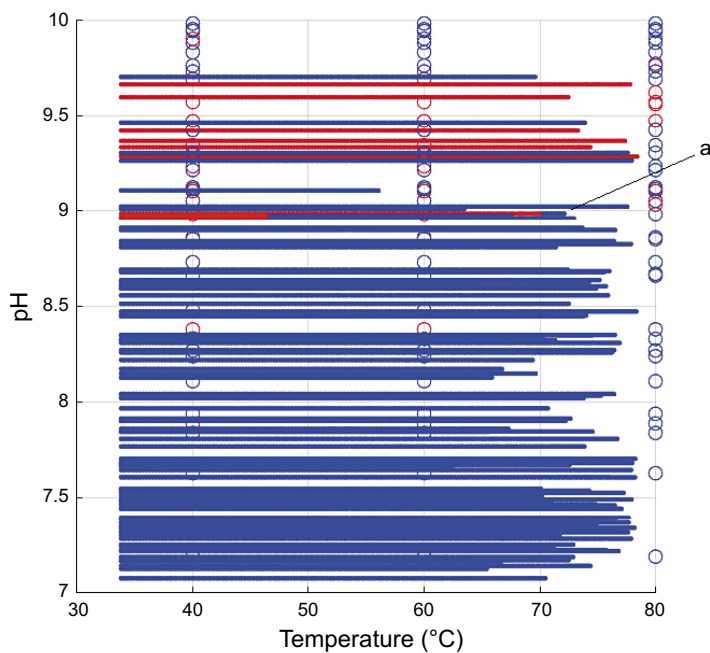


Figure 5-9. Effect of pH and temperature on active (blue)-passive (red) behaviour. The realization labelled “a” is discussed in the text. Open symbols represent the experimental data of Qin et al. shown using the same colour coding.

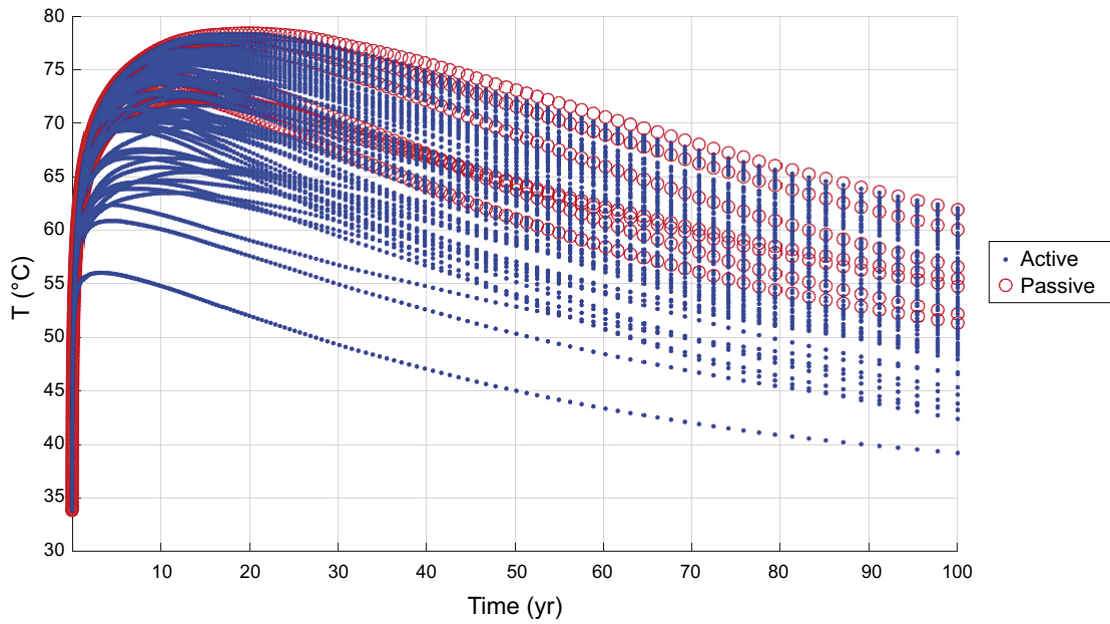


Figure 5-10. Evolution of the canister surface temperature and the predicted active (blue)-passive (red) behaviour.

Similarly, neither the sulphate (Figure 5-11) nor the bicarbonate (Figure 5-12) concentration appears to have any significant impact on whether the surface is predicted to be active or passive.

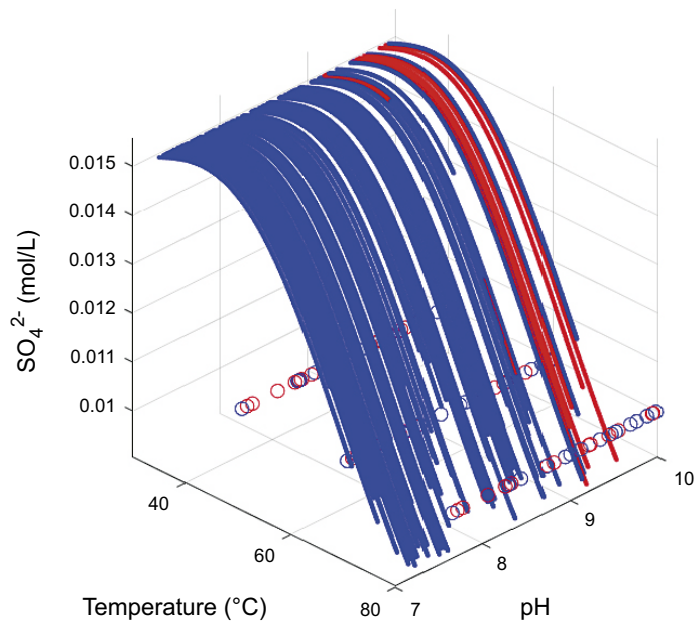


Figure 5-11. Effect of sulphate concentration, pH, and temperature on the active (blue)-passive (red) behaviour. Open symbols represent the experimental data of Qin et al. shown using the same colour coding.

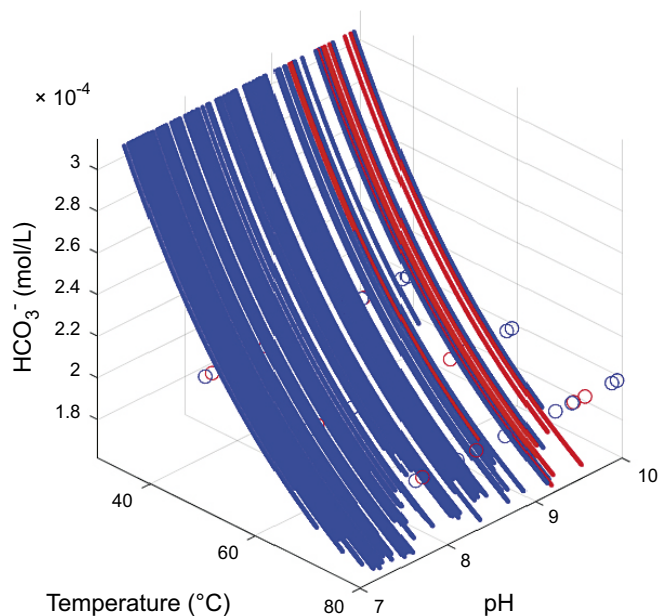


Figure 5-12. Effect of bicarbonate concentration, pH, and temperature on the active (blue)-passive (red) behaviour. Open symbols represent the experimental data of Qin et al. shown using the same colour coding.

Although passivity (as well as active behaviour) can occur throughout the 100-year simulation period, film breakdown and pitting is limited to be the first few years in the evolution of the environment. Figure 5-13 shows the time dependence of the three possible conditions for the canister surface: active, passive ($E_{\text{CORR}} < E_{\text{B}}$), or pitting ($E_{\text{CORR}} > E_{\text{RP}}$). (Also shown is the case in which film breakdown is predicted to occur, $E_{\text{CORR}} > E_{\text{B}}$, but for which there is no propagation as $E_{\text{CORR}} < E_{\text{RP}}$. This condition did not arise using the critical potential data of Qin et al. and is not discussed further here). The different lines represent individual realizations, although only a few of the 100 lines are visible as many overlap each other. For the majority of realizations, active conditions are maintained throughout the 100-year simulation period. For those few that exhibit passive behaviour (between 6 and 9 at different times for this particularly set of 100 realizations, see Figure 5-5), all are subject to film breakdown and pit initiation at early times. There then follows a period of pit growth before repassivation occurs with, in the majority of cases, the surface remaining passive for the rest of the simulation period.

There are three realizations of note in Figure 5-13. The realizations labelled “c” and “d” appear to transition from a period of film breakdown and pit growth directly to an active condition. Thus, these pits do not “repassivate” in the true sense of the term, but instead localised corrosion transitions to active dissolution because of some time-dependent change in the environment (i.e., a change in temperature, $[\text{SO}_4^{2-}]$, or $[\text{HCO}_3^-]$). The realization labelled “a” seems to be the same realization “a” identified in Figure 5-8 and Figure 5-9. For this realization, the propagating pit does repassivate (after 2 years) prior to the surface becoming first active (after 4 years) and then passive again (after ~25 years).

The early pitting behaviour is also illustrated in Figure 5-14 which shows the time dependence of E_{CORR} and of the two critical potentials E_{B} and E_{RP} . The predicted corrosion potential decreases with time because of the dependence of E_{CORR} on $[\text{O}_2]$ (Equation (3-4)) and the assumed exponential decay in $[\text{O}_2]$ with time (Equation (3-7)). The absolute value of E_{CORR} depends also on the $[\text{Cl}^-]$ selected for the particularly realization (Equation (3-4)). It is interesting to note that active realizations tend to give more-negative E_{CORR} values than passive realizations. Since the expression for E_{CORR} is based on the assumption of active conditions only (Section 3.1.4), this observed tendency is due to the fact that higher $[\text{Cl}^-]$ (and, possibly, higher T) promote active behaviour as well as more-negative E_{CORR} values. The rate of decrease of E_{CORR} with time depends on the value selected for the O_2 -consumption time constant a in each realization. Figure 5-15 shows the distribution of a values for the 100 realizations, expressed in terms of the equivalent times to reduce the initial $[\text{O}_2]$ by a factor of 99.9%. Consumption of 99.9% of the O_2 occurred within a period of approximately one year for 96 of the 100 realizations.

Finally, it is worthwhile noting that the Qin et al. critical potential data used in this simulation consistently predict E_B values greater than the E_{RP} (Figure 5-14), as would be expected based on the definition of these parameters.

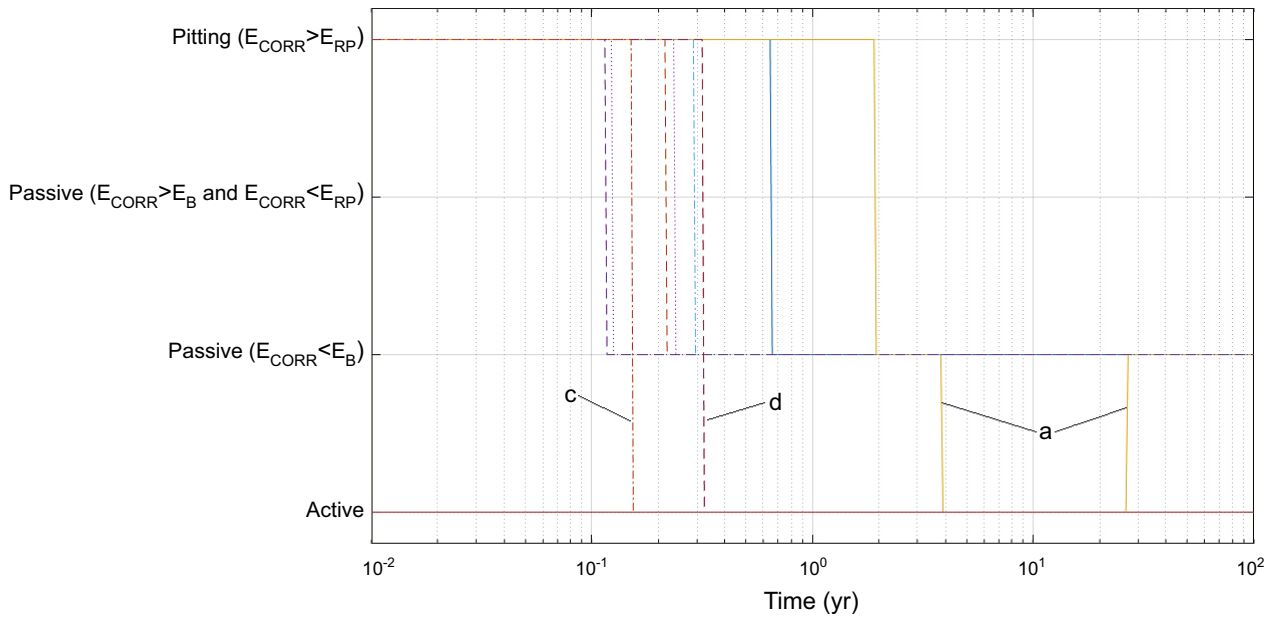


Figure 5-13. Time dependence of active conditions, film breakdown, pit propagation, and passivation. The different lines represent individual realizations.

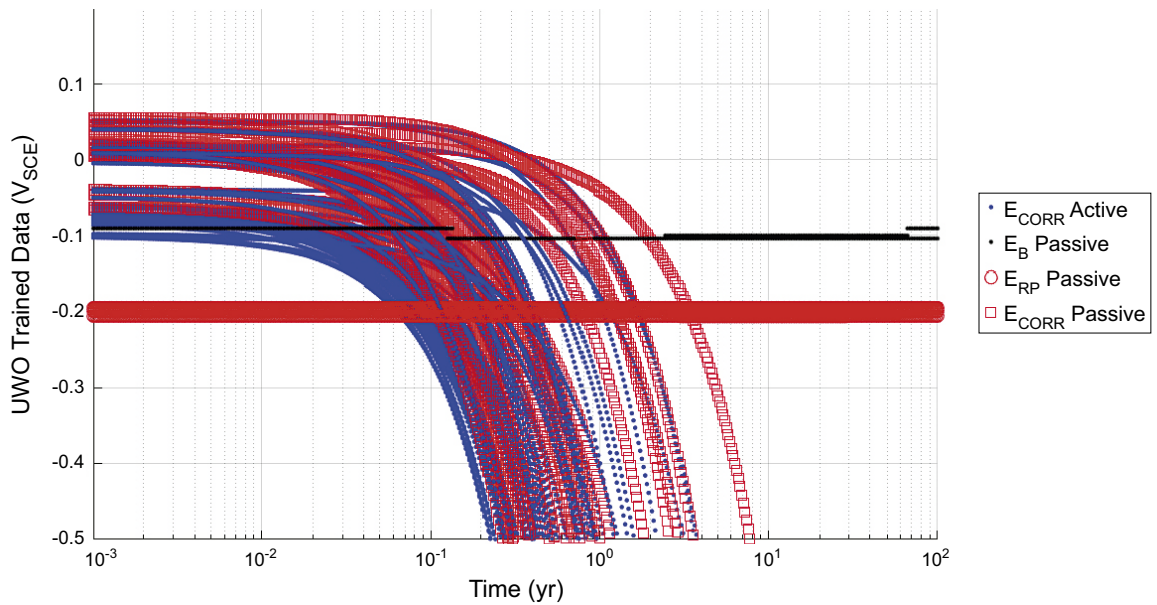


Figure 5-14. Predicted evolution of the corrosion potential E_{CORR} for active and passive conditions and of the breakdown E_B and repassivation E_{RP} potentials for passive conditions.

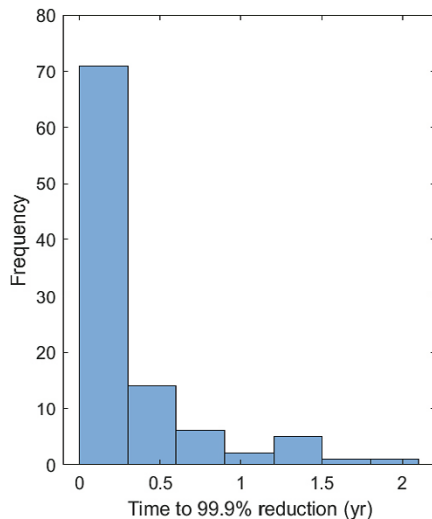


Figure 5-15. Histogram of the samples times for consumption of 99.9 % of the initial oxygen based on a simulation of 100 realizations.

5.3 Progression of pitting behaviour

As noted above, the histogram of pit depths for the full probabilistic analysis (Figure 5-2) exhibited an unusual bi-modal distribution. It is not immediately clear whether this distribution is the result of a group of pits that stop propagating at an early stage during the repository evolution in some realizations while pits continued to grow in other realizations or whether the smaller pits initiated later. In order to investigate the time dependence of pit initiation and growth in more detail, a series of 100 000-realization simulations were performed for different overall simulation times of 10^{-4} year (53 mins), 10^{-3} years (8.8 hrs), 0.01 years (3.7 days), and 0.1 years (36.5 days).

Figure 5-16 shows the pit depth distribution for each simulation time, which are also shown superimposed in Figure 5-17. From these plots, it appears that the population of very shallow pits (of the order of 10^{-8} m to 10^{-7} m in depth) does initiate at a later stage, and are not pits that initiate from the outset and re-passivate early. This second pit population appears to initiate and grow in the time period between 3.7 days and 36.5 days (0.01–0.1 years), but the pits must re-passivate relatively quickly as otherwise they would grow deeper. Figure 5-2 suggests that these pits remain even at the end of the 100-year simulation period.

These conclusions are tentative because the simulation for 0.1 years exhibited a greater degree of passivity than the other simulations (Table 5-2). As discussed in more detail in Section 5.5, this variability in predicted active-passive behaviour is to be expected even for simulations comprising 100 000 realizations, but does not seem to impact the pit depth distribution.

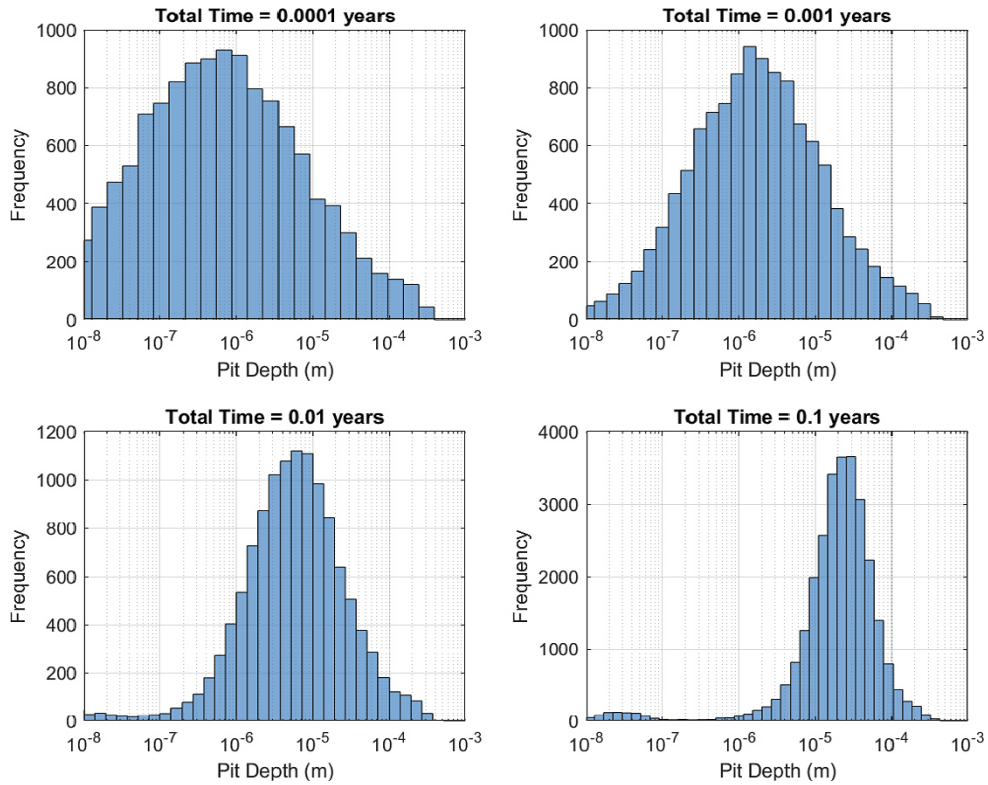


Figure 5-16. Collection of pit-depth histograms for 100 000-realization simulation periods from 10^{-4} to 0.1 years.

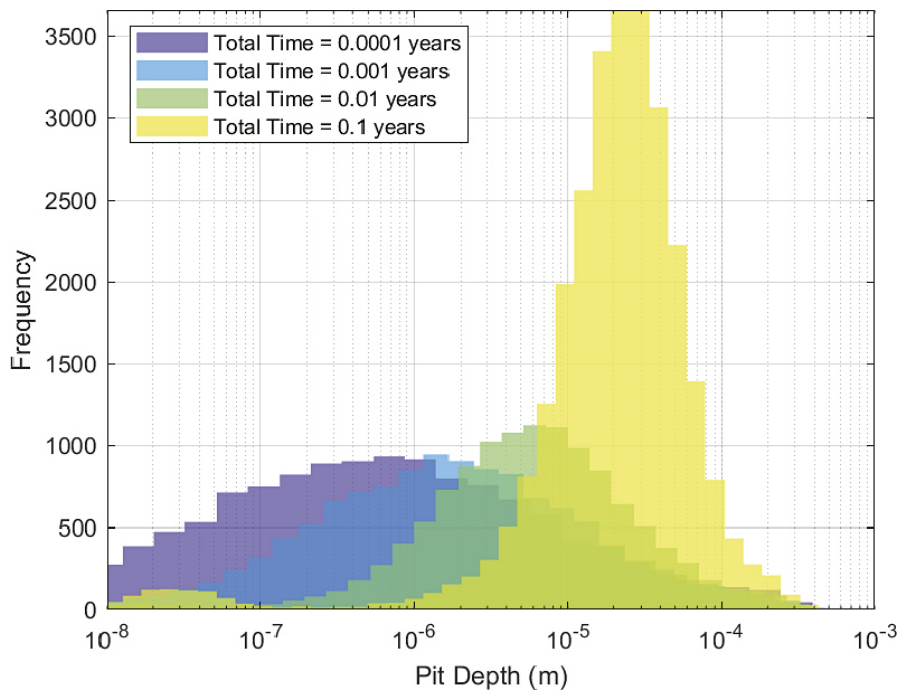


Figure 5-17. Overlay of pit-depth distributions for 100 000-realization simulation periods from 10^{-4} to 0.1 years.

Table 5-2. Summary statistics for 100 000-realization simulations with simulation periods of between 10^{-4} and 100 years.*

Simulation period	10^{-4} years	10^{-3} years	0.01 years	0.1 years	100 years
% of time steps:					
Active condition	86.8	87.9	87.9	71.9	87.2
Passive condition	13.2	12.1	12.1	28.1	12.8
% of realizations:					
Active only	86.8	87.9	87.9	71.9	86.1
Passive only	13.2	12.1	12.1	28.1	9.8
Active and passive	0.0	0.0	0.0	0.0	4.1
% of passive time steps:					
No film breakdown ($E_{\text{CORR}} < E_{\text{B}}$)	0.0	0.0	0.0	0.1	35.6
Pit initiation and growth	100.0	100.0	100.0	100.0	64.4
Film breakdown but no growth ($E_{\text{CORR}} > E_{\text{B}}$, but $E_{\text{CORR}} < E_{\text{RP}}$)	0.0	0.0	0.0	0.0	0.0
% of all realizations in which:					
Pit initiation occurred	13.2	12.1	12.1	28.1	13.3
Pit re-initiation occurred	0.0	0.0	0.0	0.0	0.0
Predicted pit depth < 100 μm	12.9	11.8	11.8	26.9	11.6
Predicted pit depth > 1 mm	0.0	0.0	0.0	0.0	0.00010
Total number of pits	13202	12107	12131	28070	13340
% of pits with depth < 100 μm	97.8	97.7	97.1	95.9	86.8
Number of pits with depth > 1 mm	0	0	0	0	1
Maximum pit depth (mm)	0.38	0.39	0.47	0.43	1.19
Mean pit depth (mm)	0.008	0.010	0.016	0.032	0.054
Pit depth standard deviation (mm)	0.030	0.031	0.033	0.035	0.054

* All simulations were performed using 100 time increments per decade starting from 10^{-5} years. The critical potential data of Qin et al. were used, along with the pit-growth kinetics for soil #65.

5.4 Effect of maximum bentonite pore-water pH

A sensitivity analysis was performed to determine the effect of the maximum value of the triangular pore-water pH distribution (Figure 3-3). As an alternative to the default value of $\text{pH}_{\text{max}} = 10$, a value of $\text{pH}_{\text{max}} = 9$ was selected as representative of more-effective pH buffering of the pore water by calcite.

Table 5-3 summarises the results of full probabilistic analyses comprising 1 000 000 realizations with the two different pH_{max} values. There is relatively little difference between the predicted behaviour in the two cases, particularly in terms of the predicted pitting behaviour. As would be expected, a greater percentage of the individual time steps and realizations exhibited active behaviour for the lower pH_{max} value, along with fewer passive conditions. The lower pH_{max} value significantly reduced the percentage of realizations exhibiting both active and passive behaviour, suggesting that the lower pH value reduces the number of realizations on the borderline between active and passive conditions.

In terms of the consequences for the predicted pitting behaviour, although there were fewer pitting events predicted at the lower pH, the size distributions of the pits were similar. Thus, the median, mean, and standard deviation of pit sizes were the same regardless of the pH_{max} value. Although the maximum pit depth was smaller for the lower pH_{max} value, there were a larger number of pits deeper than 1 mm for $\text{pH}_{\text{max}} = 9$.

Overall, therefore, there is relatively little impact of the maximum pore-water pH value on either the active-passive behaviour or the extent of pitting. This conclusion is somewhat surprising given the apparent predominant role of the pore-water pH in promoting passivation (see discussion in Section 5.2). Thus, it would seem that it is the combination of high pH and low $[\text{Cl}^-]$ that supports passivation and pit growth, not just the high pH alone.

Table 5-3. The effect of the maximum value for the triangular pH distribution for the full probabilistic analysis.*

Parameter	$\text{pH}_{\text{max}} 10$	$\text{pH}_{\text{max}} 9$
% of time steps:		
Active condition	87.2	89.6
Passive condition	12.8	10.4
% of realizations:		
Active only	83.5	88.8
Passive only	10.9	9.6
Active and passive	5.6	1.6
% of passive time steps:		
No film breakdown ($E_{\text{CORR}} < E_{\text{B}}$)	39.0	39.4
Pit initiation and growth	61.0	60.6
Film breakdown but no growth ($E_{\text{CORR}} > E_{\text{B}}$, but $E_{\text{CORR}} < E_{\text{RP}}$)	0.0	0.0
% of all realizations in which:		
Pit initiation occurred	12.4	10.4
Pit re-initiation occurred	0.0	0.0
Predicted pit depth < 100 μm	10.8	9.1
Predicted pit depth > 1 mm	0.0002	0.003
Total number of pits	124 289	104 490
Number of pits with depth < 100 μm	108 415	90 587
Number of pits with depth > 1 mm	2	3
Maximum pit depth (mm)	1.22	1.12
Median pit depth (mm)	0.039	0.039
Mean pit depth (mm)	0.053	0.054
Pit depth standard deviation (mm)	0.053	0.054

* All simulations comprised 1 000 000 realizations and were performed for a period of 100 years with 100 time increments per decade (total of 701 time increments). The critical potential data of Qin et al. were used, along with the pit-growth kinetics for soil #65.

5.5 Comparison of replicate simulations

Multiple simulations were performed for simulations comprising both 1 000 000 realizations (duplicate simulations with a maximum pH of pH 9) and 100 000 realizations (five simulations with the reference maximum pH of pH 10). Even with such large numbers of realizations, differences in the predicted active-passive and pitting behaviours are observed from simulation to simulation (Table 5-4). For example, for the five simulations with 100 000 realizations, the percentage of active conditions varies from 52.4 % to 89.4 %. The percentage of passive-only realizations (i.e., realizations for which passive conditions were maintained for the entire 100-year simulation period) varied from just 7 % to as high as 37.6 %. These differences were also observed in the pit initiation behaviour, with pitting predicted in between 11.3 % and 49.5 % of the realizations. However, there was far less difference predicted in the extent of pit growth. The fraction of pits with a depth < 100 μm only varied from 86.1 % to 87.5 % for the five replicate simulations. Although one of the five simulations predicted two pits deeper than 1 mm compared with no pits this deep in the other four simulations, the maximum pit depth for the five runs only varied from 0.69 mm to 1.09 mm. Furthermore, the median, mean, and standard deviation of the pit depth distributions were virtually identical for the five replicate simulations. The extent of pit growth depends on the pitting period and on the values of only two sampled parameters to define the pitting kinetics (A and n). Even though the number of pits is quite variable between replicate simulations (between 11 317 and 49 468), there are a sufficient number of pitting events that the variability in the factors determining pit growth is not apparent in the measures used here to characterize the pitting damage (the maximum, median, and mean pit depths).

This simulation-to-simulation variability would be expected to be reduced by increasing the number of realizations. Only two 1 000 000-realization simulations were conducted because of the length of time required to run the model (Table 5-4), so it is difficult to be certain whether the variability is reduced. In terms of the fraction of time steps exhibiting active or passive behaviour, the results from the two simulations suggest a degree of consistency, although there is still some difference in terms of the fraction of passive realizations. There does seem to be some consistency in the fraction of realizations exhibiting pit initiation as well as in the characteristics of the resulting damage, but again these conclusions are based on a sample of only two simulations.

In conclusion, it would seem that at least 1 000 000 realizations are necessary to prevent simulation-to-simulation variability, but the exact number required has not been established.

5.6 Scaling issues

In using an electrochemical approach to predicting the pitting behaviour of copper canisters, there are at least two questions related to the difference in physical size of the electrodes used for the measurements and the canisters themselves.

First, how many pits will there be on a given canister and what will be the distribution of pit depths? The main aim of the current model was to determine the effects of the variability and uncertainty in the environmental conditions in the repository on the pitting behaviour of the canisters. Accordingly, each realization represents the passivation and localised corrosion behaviour in a single deposition hole, in which the environmental conditions are implicitly assumed to be spatially uniform. Therefore, the current model does not provide any information regarding the number of pits that might occur on a given canister or the distribution of pit depths.

However, a full probabilistic analysis comprising 100 000 or 1 000 000 realizations does provide an indication of the maximum pit depth on any canister in the repository. By repeating the analysis many times, those combinations of conditions that lead to passivation, pit initiation, and an extended period of pit growth are inevitably captured. The data in Table 5-1 and Table 5-4 show that, although there is some tendency for the maximum pit depth to increase with the number of realizations, there is relatively little difference in the predicted maximum pit depth for simulations of 100 000 or more realizations. Therefore, it is reasonable to conclude that the maximum pit depth from the 1 000 000-realization simulation represents the deepest pit on any of the canisters in the repository.

Table 5-4. Comparison of the results from replicate simulations. Duplicate 1 000 000-realization simulations were run for pHmax = 9 and five 100 000-realization simulations were run for pHmax = 10.*

Number of realizations per simulation	1 000 000		100 000				
	Run 1	Run 2	Run 1	Run 2	Run 3	Run 4	Run 5
Run #							
% time steps with active condition	91.7	89.6	81.1	85.6	84.6	52.4	89.4
% time steps with passive conditions	8.3	10.4	18.9	14.4	15.4	47.6	10.6
% active only realizations	89.9	88.8	78.3	85.4	67.3	50.5	86.7
% passive only realizations	1.7	9.6	13.6	12.7	10.0	37.6	7.0
% active and passive realizations	8.4	1.6	8.1	1.9	22.7	11.9	6.4
% of passive time steps:							
No film breakdown ($E_{\text{CORR}} < E_B$)	23.8	39.4	31.1	36.0	78.5	34.8	39.0
Pit initiation and growth	76.2	60.6	68.9	64.0	51.5	65.2	61.0
Film breakdown but no growth ($E_{\text{CORR}} > E_B$, but $E_{\text{CORR}} < E_{\text{RP}}$)	0.0	0.0	0.0	0.0	0.0	0.0	0.0
% of all realizations in which:							
Pit initiation occurred	10.1	10.4	20.7	14.6	13.2	49.5	11.3
Pit re-initiation occurred	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Total number of pits	100585	104490	20731	14618	13213	49468	11317
% of pits with depth < 100 μm	87.3	86.7	87.4	87.5	86.3	87.3	86.1
Number of pits with depth > 1 mm	1	3	0	0	0	2	0
Maximum pit depth (mm)	1.00	1.12	0.86	0.69	0.88	1.09	0.93
Median pit depth (mm)	0.038	0.039	0.038	0.039	0.039	0.038	0.040
Mean pit depth (mm)	0.053	0.054	0.053	0.053	0.055	0.053	0.056
Pit depth standard deviation (mm)	0.054	0.054	0.053	0.053	0.056	0.054	0.057

* All simulations were performed for a period of 100 years with 100 time increments per decade (total of 701 time increments). The critical potential data of Qin et al. were used, along with the pit-growth kinetics for soil #65.

The second scaling question is whether the use of measurements made on electrodes of the order of a few cm² in size can be used to predict the pitting behaviour of a canister with a surface area of approximately 17 m²? For passive alloys, such as stainless steel, it has been shown that the value of E_B shifts to more-negative values with increasing electrode size (Burstein and Ilevbare 1996). This trend is associated with the increasing number of suitable initiation sites (for example, MnS inclusions) with increasing surface area. However, in the case of copper, the first requirement for pitting is that the surface be in the passive state. The results in Figure 5-1 and Table 5-1 indicate that pit initiation is predicted to occur in the majority of cases in which the surface is predicted to be passive. In other words, the probability of pitting is determined primarily by whether the surface is passive rather than by film breakdown. Thus, more-negative values of E_B due to the use of larger electrodes, or because of any other change in the experimental procedure such as the use of different potential scan rates or of different methods for pretreating the electrode, would not have significantly increased the probability of pit initiation. Since passivation, unlike pit initiation, is not a stochastic process, the size of the electrode should have little influence on the experimental conditions found to induce passivation. It is concluded, therefore, that the difference in size of the electrodes and canister has no significant impact on the predicted pitting behaviour.

The difference in size between experimental samples and the canister may also affect the pit growth kinetics. The samples used by Denison and Romanoff (1950) were of the order of 0.1–1.0 m² in size, or approximately 1 to 2 orders of magnitude smaller than the canister surface. Larger surfaces might be expected to exhibit different pit growth kinetics as there is a higher probability of a fast-growing pit on a larger surface. However, it has already been argued above that the Denison and Romanoff pit growth data are conservative as the soils were apparently aerobic with ready access of O₂. Therefore, any under-estimate of pit growth kinetics due to the small size of the experimental samples is offset by the apparently more aerobic conditions of the near-surface burial sites compared with the conditions expected for the canister surrounded by saturated compacted buffer material in a sealed deep geological repository.

6 Conclusions

A methodology has been developed for the probabilistic assessment of the pitting of copper canisters. The model has been applied to the assessment of pitting under aerobic, saturated environmental conditions. Such conditions may or may not exist during the evolution of the repository environment, depending upon the relative rates of saturation and redox evolution. Nevertheless, the techniques developed will be useful for the future assessment of the probability and extent of localised corrosion under aerobic, unsaturated conditions which will certainly exist in the repository.

The model is based on well-established electrochemical criteria for pitting corrosion and accounts for variability and uncertainty in both the pitting process and also the relevant repository environmental conditions. Data from the literature on the active-passive nature of copper surfaces and on critical potentials for passive film breakdown and repassivation are used. Environmental data representative of the near-field conditions are used to describe the evolution of the corrosive environment at the canister surface during the aerobic period.

Each model run, or realization, simulates the pitting characteristics of a canister in an individual deposition hole. In each realization, the probability of passivation is determined by comparing the selected environmental conditions with those found experimentally to induce passivity using a verified bootstrap aggregating decision tree (“bagging”) classification algorithm. The probability of pit initiation and growth is determined by comparing the value of the corrosion potential E_{CORR} to the critical potential data of Qin et al. (2017) analysed using a boosted tree regression method. Pit depths are estimated based on the predicted pitting duration and an empirical pit growth expression for an aggressive soil from the study of Denison and Romanoff (1950).

A different set of environmental conditions is defined for each realization. Values for the bentonite pore-water pH and Cl^- concentration are sampled from suitable distributions. A unique temperature-time profile is selected from a set of 6916 profiles, one for each deposition hole in the repository. The pore-water concentrations of sulphate and bicarbonate ions are calculated with the help of the PHREEQC thermodynamic code as a function of temperature based on the assumption of solubility control by the relevant solid phases. The time dependence of the dissolved O_2 concentration, required to calculate the value of E_{CORR} , is calculated on the assumption of an exponential decay, with the time required to consume 99.9 % of the initial inventory sampled from a uniform distribution of between 5 weeks and 5 years.

The full probabilistic analysis comprises one million realizations and represents an average of 145 runs per deposition hole. Time is incremented from 10^{-5} years to 100 years and at each time step the active or passive nature of the surface is determined based on the sampled and calculated environmental parameters. If the surface is passive, the probability of pit initiation is determined based on the values of E_{CORR} and of the film breakdown potential. If a pit initiates, the duration of pit propagation is estimated by comparing the value of E_{CORR} to the repassivation potential, both of which are updated at each time step as the environment evolves. A single simulation of 1 000 000 realizations with the optimised time stepping of 100 increments for each order of magnitude in time takes approximately 10 days using 20 parallel processors and 100 GB of memory.

The main aim of the work was to develop a suitable methodology for the probabilistic assessment of localised corrosion of canisters in the repository. In this regard, the MATLAB software has proven to be suitable for model development and implementation. The built-in library of statistical methods was useful for selecting and comparing different classification and regression techniques for the analysis of the active-passive and critical potential data, respectively. Other aspects of the code implementation and execution that have been optimized include:

- The time discretization required to produce a solution that is independent of the size of the time step.
- The number of realizations for which it is feasible to be able to visually discern trends and recognize patterns when plotted graphically.
- The effect of the number of realizations on the reliability of the solution.
- Simulation-to-simulation reproducibility.

Although it is recognized that aerobic, saturated conditions may or may not exist during the evolution of the repository environment, the probabilistic analysis has also provided some insight into the early evolution of the canister corrosion behaviour. In particular, the results of the simulations suggest that:

- The canister surface is more likely to be in the active, rather than the passive, condition.
- The primary factors determining active versus passive behaviour are the pore-water pH and Cl^- concentration.
- The canister temperature and the pore-water concentrations of sulphate and bicarbonate ions seem to have a minor effect on the active-passive behaviour.
- For passive conditions, pit initiation is possible but only at relatively early times when O_2 is present to sufficiently ennoble E_{CORR} .
- The duration of pitting varies from a few tens of days to a few years.
- Based on the pit-growth expressions used in the analysis, the maximum pit depth is of the order of 1 mm.
- However, the vast majority of pits are less than 100 μm deep, corresponding to the corrosion allowance for localised corrosion used for the SR-Site safety assessment.

Finally, as with all predictive models, the robustness of the predictions can be improved by improvements in the quality of the input data, especially those relating to the time-dependent evolution of the environment at the canister surface.

7 Acknowledgements

The authors thank Allan Hedin and Patrik Sellin of SKB for calculating the temperature profiles and for defining the bentonite pore-water pH and chloride ion concentrations, respectively.

References

SKB's (Svensk Kärnbränslehantering AB) publications can be found at www.skb.com/publications.

- Battino R, Rettich T R, Tominaga T, 1983.** The solubility of oxygen and ozone in liquids. *Journal of Physical and Chemical Reference Data* 12, 163–178.
- Birgersson M, Karnland O, 2009.** Ion equilibrium between montmorillonite interlayer space and an external solution – Consequences for diffusional transport. *Geochimica et Cosmochimica Acta* 73, 1908–1923.
- Breiman L, 1996.** Bagging predictors. *Machine Learning* 24, 123–140.
- Burstein G T, Ilevbare G O, 1996.** The effect of specimen size on the measured pitting potential of stainless steel. *Corrosion Science* 38, 2257–2265.
- Cong H, Michels H T, Scully J R, 2009.** Passivity and pit stability behaviour of copper as a function of selected water chemistry variables. *Journal of The Electrochemical Society* 156, C16–C27.
- Denison I A, Romanoff M, 1950.** Soil-corrosion studies, 1946 and 1948: copper alloys, lead, and zinc. *Journal of research of the National Bureau of Standards* 44, 259–289.
- Frankel G S, Sridhar N, 2008.** Understanding localized corrosion. *Materials Today* 11, 38–44.
- Friedman J H, 2001.** Greedy function approximation: a gradient boosting machine. *The Annals of Statistics* 29, 1189–1232.
- Giroud N, Tomonaga Y, Wersin P, Briggs S, King F, Diomidis N, 2018.** On the fate of oxygen in a spent fuel emplacement drift in Opalinus Clay. *Applied Geochemistry* 97, 270–278.
- Hedin A, 2004.** Integrated near-field evolution model for a KBS-3 repository. SKB R-04-36, Svensk Kärnbränslehantering AB.
- Imai H, Fukuda T, Akashi M, 1996.** Effects of anionic species on the polarization behaviour of copper for waste package material in artificial ground water. *MRS Online Proceedings Library* 412, 589–596.
- Johansson A J, Lilja C, Hedin A, 2019.** Corrosion of copper in repository-like field tests: compilation and analysis of data. In *Proceedings of International High-level Radioactive Waste Management 2019*, Knoxville, TN, 14–18 April 2019. La Grange Park, IL: American Nuclear Society, 316–327.
- King F, Lilja C, 2013.** Localised corrosion of copper canisters in bentonite pore water. SKB TR-13-27, Svensk Kärnbränslehantering AB.
- King F, Lilja C, 2014.** Localised corrosion of copper canisters. *Corrosion Engineering, Science and Technology* 49, 420–424.
- King F, Litke C D, Quinn M J, LeNeveu D M, 1995.** The measurement and prediction of the corrosion potential of copper in chloride solutions as a function of oxygen concentration and mass-transfer coefficient. *Corrosion Science* 37, 833–851.
- King F, Kolář M, Maak P, 2008.** Reactive-transport model for the prediction of the uniform corrosion behaviour of copper used fuel containers. *Journal of Nuclear Materials* 379, 133–141.
- King F, Lilja C, Pedersen K, Pitkänen P, Vähänen M, 2010.** An update of the state-of-the-art report on the corrosion of copper under expected conditions in a deep geologic repository. SKB TR-10-67, Svensk Kärnbränslehantering AB.
- King F, Kolář M, Vähänen M, 2011a.** Reactive-transport modelling of the sulphide-assisted corrosion of copper nuclear waste containers. In Kursten B, Feron D, Druyts F (eds). *Sulphur-assisted corrosion in nuclear disposal systems*. Leeds, UK: Maney Publishing. (European Federation of Corrosion Publications 59) 152–164.
- King F, Kolář M, Vähänen M, Lilja C, 2011b.** Modelling the long-term corrosion behaviour of copper canisters in a KBS-3 repository. *Corrosion Engineering, Science and Technology* 46, 217–222.

- King F, Hall D S, Keech P G, 2017.** Nature of the near-field environment in a deep geological repository and the implications for the corrosion behaviour of the container. *Corrosion Engineering, Science and Technology* 52, sup1, 25–30.
- Kong D, Dong C, Xu A, Man C, He C, Li X, 2017a.** Effect of sulfide concentration on copper corrosion in anoxic chloride-containing solutions. *Journal of Materials Engineering and Performance* 26, 1741–1750.
- Kong D, Xu S, Dong C, Mao F, Xiao K, Li X, Macdonald D D, 2017b.** Electrochemical investigation and *ab initio* computation of passive film properties on copper in anaerobic sulphide solutions. *Corrosion Science* 116, 34–43.
- Macdonald D D, Mao F, Dong C, Sharifi-Asl S, 2016.** Measurement of parameter values for predicting corrosion phenomena on copper in Swedish HLNW repositories. Phase IV: impact of chloride ion on the passivity and pitting of copper. Report 2016:30, Swedish Radiation Safety Authority.
- Mao F, Dong C, Sharifi-Asl S, Lu P, Macdonald D D, 2014.** Passivity breakdown on copper: influence of chloride ion. *Electrochimica Acta* 144, 391–399.
- MathWorks, 2018.** MATLAB Release R2018b. Natick, MA, The MathWorks, Inc.
- Muurinen A, Carlsson T, 2008.** Eh and pH in compacted MX-80 bentonite. Posiva Working Report 2008-48, Posiva Oy, Finland.
- Olsson S, Jensen V, Johannesson L-E, Hansen E, Karnland O, Kumpulainen S, Svensson D, Hansen S, Lindén J, 2013.** Prototype Repository. Hydro-mechanical, chemical and mineralogical characterization of the buffer and backfill material from the outer section of the Prototype Repository. SKB TR-13-21, Svensk Kärnbränslehantering AB.
- Puigdomenech I, Taxén C, 2000.** Thermodynamic data for copper. Implications for the corrosion of copper under repository conditions. SKB TR-00-13, Svensk Kärnbränslehantering AB.
- Puigdomenech I, Ambrosi J-P, Eisenlohr L, Lartigue J-E, Banwart S A, Bateman K, Milodowski A E, West J M, Griffault L, Gustafsson E, Hama K, Yoshida H, Kotelnikova S, Pedersen K, Michaud V, Trotignon L, Rivas Perez J, Tullborg E-L, 2001.** O₂ depletion in granitic media. The REX project. SKB TR-01-05, Svensk Kärnbränslehantering AB.
- Qin Z, Daljeet R, Ai M, Harhangi N, Noël J J, Ramamurthy S, Shoesmith D, King F, Keech P, 2017.** The active/passive conditions for copper corrosion under nuclear waste repository environment. *Corrosion Engineering, Science and Technology* 52, sup1, 45–49.
- Romanoff M, 1989.** Underground corrosion. Houston, TX: NACE International.
- Scully J R, Edwards M, 2013.** Review of the NWMO copper corrosion allowance. NWMO TR-2013-04, Nuclear Waste Management Organization, Canada
- Scully J R, Hicks T W, 2012.** Initial review phase for SKB's safety assessment SR-Site: corrosion of copper. Technical Note 2012:21, Swedish Radiation Safety Authority.
- Scully J R, Féron D, Hänninen H, 2016.** Review of the NWMO copper corrosion program. NWMO TR-2016-11, Nuclear Waste Management Organization, Canada.
- Sena C, Salas J, Arcos D, 2010.** Aspects of geochemical evolution of the SKB near field in the frame of SR-Site. SKB TR-10-59, Svensk Kärnbränslehantering AB.
- SKB, 2010.** Corrosion calculations report for the safety assessment SR-Site. SKB TR-10-66, Svensk Kärnbränslehantering AB.
- SKBF/KBS, 1978.** Copper as a canister material for unprocessed nuclear waste – evaluation with respect to corrosion. Final report. KBS TR 90, SKBF/KBS Svensk Kärnbränslehantering AB.
- Soltis J, 2015.** Passivity breakdown, pit initiation and propagation of pits in metallic materials – Review. *Corrosion Science* 90, 5–22.
- Sridhar N, Cragolino G A, 1993.** Applicability of repassivation potential for long-term prediction of localized corrosion of Alloy 825 and Type 316L stainless steel. *Corrosion* 49, 885–894.
- Szklarska-Smialowska Z, 2005.** Pitting and crevice corrosion. Houston, TX: NACE International.

Thompson N G , Syrett B C, 1992. Relationship between conventional pitting and protection potentials and a new, unique pitting potential. *Corrosion* 48, 649–659.

Vapnik V N, 1995. The nature of statistical learning theory. New York: Springer.

Wanner H, Wersin P, Sierro N, 1992. Thermodynamic modelling of bentonite-groundwater interaction and implications for near field chemistry in a repository for spent fuel. SKB TR 92-37, Svensk Kärnbränslehantering AB.

Source Code

A1 Source code

A1.1 Localised corrosion model

A statistical approach to the localised corrosion of copper containers. At each time step, we track the pit growth. The status of the pit is -20 at initialization. Meaning no calculation took place. If the surface is active, it results in a pit value of -15 (no pitting). If it is passive, a pit may initiate. If $E_{CORR} < E_B$ then return to main control with -10 pitgrowth. If $E_{CORR} > E_B$ then the pit will propagate. The pit will continue to propagate as long as $E_{CORR} > E_{RP}$ (-5 if $E_{CORR} < E_{RP}$) (time is tracked and is used to return total piting $+1$ for each timestep).

```
%The code can be run as a MATLAB script or called as a function
%function
localizedCorrosionCPU(time,logtimestep,KNN_num,mc,RUN,PitGrowthExp,PotentialExp,ClassifierMethod,RegressMethod_Eb,RegressMethod_Erp)
```

A1.2 Program initialization

Clear variable space and close any open MATLAB figures to ensure a clean workspace.

```
%
%Comment out this block if running as function
%

clearvars;
close all;

%
%Comment out this block if running as function
%

% initialize the random number generator in a new location
rng('shuffle');
% for replication initiate the RNG in the same location
% rng(1,'twister');
```

A1.3 Program constants and input initialization

Define constants and initialization of statistical variables.

```
tic; %performance timing

%
%Comment out this block if running as function
%

RUN = '\Run 1\'; %to differentiate duplicate model runs
mc = 100; %number of model realizations to run
time = 100; %MIN: 1e-5 years; MAX 10,000 years (currently)
```

```

logtimestep = 10; %time steps in log time in increments 1/logtimestep
KNN_num = 10; %number of KNN nearest neighbours
                %if running ClassifierMethod = 0 below

%Pit Growth Expression
PitGrowthExp = 1; % 0 -- constant: 5e-6 m/year
                 % 1 -- Denison and Romanoff (1950) #65 Chino soil
                 % 2 -- Denison and Romanoff (1950) #66 Mohave soil

%Method for evaluating Eb and Erb potentials
PotentialExp = 1; % 0 -- Cong and Scully expressions for Eb and Erp
                 % 1 -- Training on UWO data

%Method for Active/Passive data classification
ClassifierMethod = 2; % 0 -- original KNN implementation
                    % 1 -- Bagged Tree
                    % 2 -- Bagged Tree (Verified)
                    % 3 -- KNN Fine
                    % 4 -- KNN Fine (Verified)
                    % 5 -- KNN Medium
                    % 6 -- KNN Medium (Verified)
                    % 7 -- KNN Weighted
                    % 8 -- KNN Weighted (Verified)

RegressMethod_Eb = 4; % 0 -- original regression implementation
                    % 1 -- Bagged Tree
                    % 2 -- Bagged Tree (Verified)
                    % 3 -- Boosted Tree
                    % 4 -- Boosted Tree (Verified)
                    % 5 -- Complex Tree
                    % 6 -- Complex Tree (Verified)
                    % 7 -- FG SVM
                    % 8 -- FG SVM (Verified)
                    % 9 -- GPR Exp
                    %10 -- GPR Exp (Verified)
                    %11 -- Medium Tree
                    %12 -- Medium Tree (Verified)

RegressMethod_Erp = 4; % 0 -- original regression implementation
                    % 1 -- Bagged Tree
                    % 2 -- Bagged Tree (Verified)
                    % 3 -- Boosted Tree
                    % 4 -- Boosted Tree (Verified)
                    % 5 -- Complex Tree
                    % 6 -- Complex Tree (Verified)
                    % 7 -- FG SVM
                    % 8 -- FG SVM (Verified)
                    % 9 -- GPR Exp
                    %10 -- GPR Exp (Verified)
                    %11 -- Medium Tree
                    %12 -- Medium Tree (Verified)

%
%Comment out this block if running as function
%

%figures can become compute intensive to draw if mc is too large
if mc < 200
    INPUT_FIGURES = 1; % 1 to show figures; 0 to hide

```

```

AP_FIGURES = 1;
PRINT_FIGURES = 1;
else
    INPUT_FIGURES = 0; % 1 to show figures; 0 to hide
    AP_FIGURES = 0;
    PRINT_FIGURES = 0;
end

%use a logarithmically spaced time step
time_scale = 10.^((log10(1.0e-5):1/logtimestep:log10(time)));

%initialize variables
PitGrowth = zeros(mc,length(time_scale)) - 20;
PitTime = zeros(mc,length(time_scale));
SO4 = zeros(mc,length(time_scale));
HCO3 = zeros(mc,length(time_scale));
CL = zeros(mc,1);

%specify a maximum corrosion allowance for each realization
CorrosionAllowance = 0.05; %meters

%Ecorr Calculation constants Eq 14 from King et al. 1995
F = 96487; %C/mol Faraday constant
R = 8.314; %J/K/mol Gas Constant
n_a = 1; %number of anodic electrons
n_c = 4; %number of cathodic electrons
E_a_0 = -0.105; %V(SCE) standard potential
k_a = 3.3e-4; %dm4/mol/s
k_2_r = 1.42e-3; %dm/s
D_O2 = 1.7e-5; %cm2/s
D_CuCl2 = 5.5e-6; %cm2/s

%output folder location
%the subfolder must exist or an error will be thrown
subfolder = strcat('\Results\', 'time=', num2str(time), '\');
filename = strcat('mc', num2str(mc), ' time', num2str(time), ...
    ' logtimestep', num2str(logtimestep), ...
    ' Pit', num2str(PitGrowthExp), ...
    ' Pot', num2str(PotentialExp), ...
    ' ClaM', num2str(ClassifierMethod), ...
    ' ReM_Eb', num2str(RegressMethod_Eb), ...
    ' ReM_Erb', num2str(RegressMethod_Erp), ' Run', RUN(6));

```

A1.4 Temperature data input

Two temperature data input types 1. Data from the entire repository a) All Dry b) All Wet 2. Data of three profiles and interpolated using a triangular distribution.

```

%1. Data from the entire repository
%temperatureFile = 'TemperatureAllDry.xlsx'; extrap_temp = 54.247;
temperatureFile = 'TemperatureAllWet.xlsx'; extrap_temp = 33.869;
rangeT = 'A2:AY6917';
sheetT = 1;
%import data
T_Data = xlsread(temperatureFile, sheetT, rangeT);
T_Data_times = xlsread(temperatureFile, sheetT, 'A1:AY1'); %years

```

```

%interpolate temperature based on defined time_scale
querypoints = time_scale;
interpTemp = zeros(length(T_Data),length(time_scale));
for i = 1:length(T_Data)
    interpTemp(i,:) = interp1(T_Data_times, T_Data(i,:), ...
        querypoints,'pchip',extrap_temp);
end
%randomly sample a temperature profile for each realization
T = zeros(mc,length(time_scale),'double');
Tsample = unidrnd(length(T_Data),[mc,1]); %samples to choose.
%from all interpolated temperature profiles, choose the samples
%instead of a loop over 1:1:mc, use vector form:
T(1:1:mc,:) = interpTemp(Tsample(1:1:mc),:);

% 2. Data from three profiles (this method has not been tested recently)
%% spreadsheet data range and sheet number
% rangeT = 'B5:E26';
% sheetT = 1;
%
% temperatureFile = 'TemperatureTD.xlsx';
%% for a single spreadsheet data import
% T_Data = importfileTemperatureTD(temperatureFile,sheetT,rangeT);
%% First interpolate temperature data
% querypoints = time_scale;
% interpLOW = interp1(T_Data.Time,T_Data.LOW,querypoints,'pchip',48.715);
% interpMED = interp1(T_Data.Time,T_Data.MEDIUM,querypoints,'pchip',69.013);
% interpHIGH = interp1(T_Data.Time,T_Data.HIGH,querypoints,'pchip',72.212);
% T = zeros(mc,time);
% Tsample = rand(mc,1); %percentiles to choose.
% for i = length(time_scale)
%     %independent random deviates each time and mc
%     %pdT(i) = makedist('Triangular', ...
%     %     'a',interpLOW(i),'b',interpMED(i),'c',interpHIGH(i));
%     %T(:,i) = random(pdT(i),mc,1); %temperature
%
%     %choose a specific random deviate and maintain it throughout time
%     %but use different percentiles each mc
%     a = interpLOW(i);
%     b = interpHIGH(i);
%     c = interpMED(i);
%     f = (c-a)/(b-a);
%     for j = 1:1:mc
%         if Tsample(j) < f
%             T(j,i) = a+sqrt(Tsample(j)*(b-a)*(c-a));
%         else
%             T(j,i) = b-sqrt((1-Tsample(j))*(b-a)*(b-c));
%         end
%     end
% end

if INPUT_FIGURES
    f1 = figure;
    ax1 = axes('Parent',f1);
    xscale = time_scale;
    log_xscale = [1e-5 time];
    plot(xscale,T','color',[0.2 0.2 0.2 0.2],'Parent',ax1);
    xlabel('Time (yr)'); ylabel('Temperature (\circ C)'); grid on;

```



```

f2 = figure;
ax2 = axes('Parent',f2);
semilogx(xscale,T','color',[0.2 0.2 0.2 0.2],'Parent',ax2);
xlabel('Time (yr)'); ylabel('Temperature (\circC)'); grid on;
xlim(log_xscale);

if PRINT_FIGURES
    saveas(f1,[subfolder filename ' 01 ' temperatureFile(1:end-5) ' linear'
], 'png');
    saveas(f2,[subfolder filename ' 02 ' temperatureFile(1:end-5) ' semilog'
], 'png');
end
end
end

```

A1.5 Active/passive data input

Import spreadsheet using function “importfile.m” generated from built-in MATLAB Import Data Tool.

```

rangeAP = 'A5:Q628'; %APData02.xlsx
sheetAP = 1;
% spreadsheet data import
APData = importfile('APData02.xlsx',sheetAP,rangeAP);

%response data for fitting
active = APData.active; %1 for active; 0 for passive
%Data for fitting (use concentrations, ph and temp)
apdata = table( APData.C1, APData.SO4, APData.HCO3, ...
                APData.pH, APData.T,...
                'variableNames',{'C1','SO4','HCO3','pH','T'});
%apply selected classification method
switch ClassifierMethod
case 0 %original KNN implementation
    %Use knn algorithm
    trainedClassifier = fitcknn(apdata,active,'NumNeighbors',KNN_num, ...
                                'Standardize',1, ...
                                'Distance','euclidean');
    Md2 = ExhaustiveSearcher(table2array(apdata));
case 1 % Bagged Tree
    trainingData = table( APData.C1, APData.SO4, APData.HCO3, ...
                          APData.pH, APData.T, APData.active,...
                          'variableNames',{'C1','SO4','HCO3','pH','T','active'});
    [trainedClassifier, validationAccuracy] =
trainClassifier_Bagged_Tree(trainingData);
case 2 % Bagged Tree (Verified)
    trainingData = table( APData.C1, APData.SO4, APData.HCO3, ...
                          APData.pH, APData.T, APData.active,...
                          'variableNames',{'C1','SO4','HCO3','pH','T','active'});
    [trainedClassifier, validationAccuracy] =
trainClassifier_Bagged_Tree_Ver(trainingData)
case 3 % KNN Fine
    trainingData = table( APData.C1, APData.SO4, APData.HCO3, ...
                          APData.pH, APData.T, APData.active,...
                          'variableNames',{'C1','SO4','HCO3','pH','T','active'});
    [trainedClassifier, validationAccuracy] =
trainClassifier_KNN_Fine(trainingData);

```

```

case 4 % KNN Fine (Verified)
    trainingData = table( APData.C1, APData.S04, APData.HCO3, ...
        APData.pH, APData.T, APData.active,...
        'VariableNames',{'C1','S04','HCO3','pH','T','active'});
    [trainedClassifier, validationAccuracy] =
trainClassifier_KNN_Fine_Ver(trainingData);
case 5 % KNN Med
    trainingData = table( APData.C1, APData.S04, APData.HCO3, ...
        APData.pH, APData.T, APData.active,...
        'VariableNames',{'C1','S04','HCO3','pH','T','active'});
    [trainedClassifier, validationAccuracy] =
trainClassifier_KNN_Medium(trainingData);
case 6 % KNN Med (Verified)
    trainingData = table( APData.C1, APData.S04, APData.HCO3, ...
        APData.pH, APData.T, APData.active,...
        'VariableNames',{'C1','S04','HCO3','pH','T','active'});
    [trainedClassifier, validationAccuracy] =
trainClassifier_KNN_Medium_Ver(trainingData);
case 7 % KNN Weighted
    trainingData = table( APData.C1, APData.S04, APData.HCO3, ...
        APData.pH, APData.T, APData.active,...
        'VariableNames',{'C1','S04','HCO3','pH','T','active'});
    [trainedClassifier, validationAccuracy] =
trainClassifier_KNN_Weighted(trainingData);
case 8 % KNN Weighted (Verified)
    trainingData = table( APData.C1, APData.S04, APData.HCO3, ...
        APData.pH, APData.T, APData.active,...
        'VariableNames',{'C1','S04','HCO3','pH','T','active'});
    [trainedClassifier, validationAccuracy] =
trainClassifier_KNN_Weighted_Ver(trainingData);
end

%Data for Eb and Erp fitting (use concentrations, ph and temp)
ebdata_raw = table( APData.C1, APData.S04, APData.HCO3, ...
    APData.pH, APData.T, APData.Eb, ...
    'VariableNames',{'C1','S04','HCO3','pH','T','Eb'});
%remove NaN values
ebdata_nonan = ebdata_raw(all(~isnan(table2array(ebdata_raw)),2),:);

erpdata_raw = table( APData.C1, APData.S04, APData.HCO3,...
    APData.pH, APData.T, APData.Erp, ...
    'VariableNames',{'C1','S04','HCO3','pH','T','Erp'});
%remove NaN values
erpdata_nonan = erpdata_raw(all(~isnan(table2array(erpdata_raw)),2),:);

%train selected regression method
switch RegressMethod_Eb
case 0
    [trainedModel_Eb, validationRMSE_Eb] =
trainRegressionModel_Eb_FG_SVM(ebdata_nonan);
case 1
    [trainedModel_Eb, validationRMSE_Eb] =
trainRegressionModel_Eb_Bagged_Tree(ebdata_nonan);
case 2
    [trainedModel_Eb, validationRMSE_Eb] =
trainRegressionModel_Eb_Bagged_Tree_Ver(ebdata_nonan);
case 3
    [trainedModel_Eb, validationRMSE_Eb] =

```

```

trainRegressionModel_Eb_Boosted_Tree(ebdata_nonan);
    case 4
        [trainedModel_Eb, validationRMSE_Eb] =
trainRegressionModel_Eb_Boosted_Tree_Ver(ebdata_nonan)
    case 5
        [trainedModel_Eb, validationRMSE_Eb] =
trainRegressionModel_Eb_Complex_Tree(ebdata_nonan);
    case 6
        [trainedModel_Eb, validationRMSE_Eb] =
trainRegressionModel_Eb_Complex_Tree_Ver(ebdata_nonan);
    case 7
        [trainedModel_Eb, validationRMSE_Eb] =
trainRegressionModel_Eb_FG_SVM(ebdata_nonan);
    case 8
        [trainedModel_Eb, validationRMSE_Eb] =
trainRegressionModel_Eb_FG_SVM_Ver(ebdata_nonan);
    case 9
        [trainedModel_Eb, validationRMSE_Eb] =
trainRegressionModel_Eb_GPR_Exp(ebdata_nonan);
    case 10
        [trainedModel_Eb, validationRMSE_Eb] =
trainRegressionModel_Eb_GPR_Exp_Ver(ebdata_nonan);
    case 11
        [trainedModel_Eb, validationRMSE_Eb] =
trainRegressionModel_Eb_Medium_Tree(ebdata_nonan);
    case 12
        [trainedModel_Eb, validationRMSE_Eb] =
trainRegressionModel_Eb_Medium_Tree_Ver(ebdata_nonan);
end

switch RegressMethod_Erp
    case 0
        [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_FG_SVM(erpdata_nonan);
    case 1
        [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_Bagged_Tree(erpdata_nonan);
    case 2
        [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_Bagged_Tree_Ver(erpdata_nonan);
    case 3
        [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_Boosted_Tree(erpdata_nonan);
    case 4
        [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_Boosted_Tree_Ver(erpdata_nonan)
    case 5
        [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_Complex_Tree(erpdata_nonan);
    case 6
        [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_Complex_Tree_Ver(erpdata_nonan);
    case 7
        [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_FG_SVM(erpdata_nonan);
    case 8
        [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_FG_SVM_Ver(erpdata_nonan);

```

```

case 9
    [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_GPR_Exp(erpdata_nonan);
case 10
    [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_GPR_Exp_Ver(erpdata_nonan);
case 11
    [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_Medium_Tree(erpdata_nonan);
case 12
    [trainedModel_Erp, validationRMSE_Erp] =
trainRegressionModel_Erp_Medium_Tree_Ver(erpdata_nonan);
end

```

A1.6 Import chloride concentration data

Data is assumed to be in mol/l Import spreadsheet using function “importChlorideCDF.m” generated from built-in MATLAB Import Data Tool.

```

% spreadsheet data range and sheet number
rangeT = 'A2:D44';
sheetT = 1;

% spreadsheet data import
ChlorideCDF = importChlorideCDF('ChlorideCDF.xlsx',sheetT,rangeT);

%take from table input the chloride concentration data
cdf = ChlorideCDF.Clmol; %Chloride concentration [mol/l]

%fit a CDF to the data
[percentile,cl_conc,~,~] = ecdf(cdf);

%add interpolation data to dataset
percentile_interp = 0:0.001:1;
CL_cdf = interp1(percentile,cl_conc,percentile_interp);

%choose chloride concentration random percentiles
clsample = rand(mc,1); %percentiles to choose.
for i = 1:1:mc
    %choose the chloride sample closest to each random percentile
    %1. filter all values less than clsample
    %2. then take the max
    %3. the results is the array value closest to clsample (from the left)
    CL(i) = max(CL_cdf(percentile_interp<=clsample(i)));
end

if INPUT_FIGURES
    f3 = figure;
    ax3 = axes('Parent', f3);
    cdfplot(cdf);
    hold on;
    %plot(cl_conc,percentile)
    plot(CL_cdf,percentile_interp,'Parent',ax3)
    hold off;
    title(''); xlabel('Cl^{i} (mol/L)'); ylabel('CDF - Probability');
    legend('Chloride Input Data','Interpolated Chloride Data','Location','southeast')
end

```

```

f4 = figure;
ax4 = axes('Parent', f4);
histogram(CL, 'Parent', ax4);
xlabel('Cl- (mol/L)'); ylabel('Frequency');

if PRINT_FIGURES
    saveas(f3,[subfolder filename ' 03 Chloride CDF' ],'png');
    saveas(f4,[subfolder filename ' 04 Chloride Histogram' ],'png');
end
end

```

A1.7 Generate environmental parameters

Determine RH, pH, SO₄²⁻, HCO₃⁻, O₂.

```

%initialize a triangular probability distribution for pH
pHmin = 7.0;
pHpeak = 7.4;
pHmax = 10.0;
pdpH = makedist('Triangular','a',pHmin,'b',pHpeak,'c',pHmax);
%randomly select pH for each realization
pH = random(pdpH,mc,1);
%calculate OH from pH
OH = 1e-14 * 10.^pH;

if INPUT_FIGURES
    f5 = figure;
    ax5 = axes('Parent', f5);
    plotrange = pHmin: 0.001 : pHmax;
    plot(plotrange,pdf(pdpH,plotrange), 'Parent', ax5)
    xlabel('pH'); ylabel('Probability');

    f6 = figure;
    ax6 = axes('Parent', f6);
    yyaxis left
    histogram(pH, 'Parent', ax6);
    xlabel('pH'); ylabel('Frequency');
    hold on;
    yyaxis right
    plot(plotrange,pdf(pdpH,plotrange), 'Linewidth', 2, 'Parent', ax6)
    ylabel('Probability');

    if PRINT_FIGURES
        saveas(f5,[subfolder filename ' 05 pH Distribution' ],'png');
        saveas(f6,[subfolder filename ' 06 pH Histogram' ],'png');
    end
end

%Generate SO4 and HCO3 profiles as a function of temperature
%instead of a loop over 1:1:mc, use vector form
i = 1:1:mc;
SO4(i,:) = - 1.26901382E-12 * T(i,:).^6 + 3.97938117E-10 * T(i,:).^5 ...
            - 4.74084833E-8 * T(i,:).^4 + 2.68766739E-6 * T(i,:).^3 ...
            - 7.73780299E-5 * T(i,:).^2 + 1.13158668E-3 * T(i,:) ...
            + 7.89962675E-3;
HCO3(i,:) = - 5.10348861E-10 * T(i,:).^3 + 1.33530950E-7 * T(i,:).^2 ...

```

```

- 1.33530950E-5 * T(i,:) + 6.33530950E-4;

if INPUT_FIGURES
    f7 = figure;
    ax7a = axes('Parent',f7);
    ax7b = axes('Parent',f7);
    subplot(1,2,1,ax7a);
    f7tr = 10:1:95;
    plot(f7tr,- 1.26901382E-12 * f7tr.^6 + 3.97938117E-10 * f7tr.^5 ...
        - 4.74084833E-8 * f7tr.^4 + 2.68766739E-6 * f7tr.^3 ...
        - 7.73780299E-5 * f7tr.^2 + 1.13158668E-3 * f7tr ...
        + 7.89962675E-3, 'Parent', ax7a)
    xlabel('Temperature (\circC)'); ylabel('SO_{4} (mol/L)'); grid on;
    subplot(1,2,2,ax7b);
    plot(f7tr,- 5.10348861E-10 * f7tr.^3 + 1.33530950E-7 * f7tr.^2 ...
        - 1.33530950E-5 * f7tr + 6.33530950E-4, 'Parent', ax7b)
    xlabel('Temperature (\circC)'); ylabel('HCO_{3} (mol/L)'); grid on;
    pos = get(f7,'position');
    set(f7,'position',[pos(1) pos(2) pos(3)*1.5 pos(4)]);

    f8 = figure;
    ax8a = axes('Parent',f8);
    ax8b = axes('Parent',f8);
    subplot(1,2,1,ax8a)
    plot(xscale,S04,'color',[0.2 0.2 0.2 0.2], 'Parent', ax8a);
    xlabel('Time (yr)'); ylabel('SO_{4} (mol/L)'); grid on;
    subplot(1,2,2,ax8b)
    semilogx(xscale,S04,'color',[0.2 0.2 0.2 0.2], 'Parent', ax8b);
    xlabel('Time (yr)'); ylabel('SO_{4} (mol/L)'); grid on;
    xlim(log_xscale);
    pos = get(f8,'position');
    set(f8,'position',[pos(1) pos(2) pos(3)*1.5 pos(4)]);

    f9 = figure;
    ax9a = axes('Parent',f9);
    ax9b = axes('Parent',f9);
    subplot(1,2,1,ax9a)
    plot(xscale,HCO3,'color',[0.2 0.2 0.2 0.2], 'Parent', ax9a);
    xlabel('Time (yr)'); ylabel('HCO_{3} (mol/L)'); grid on;
    subplot(1,2,2,ax9b)
    semilogx(xscale,HCO3,'color',[0.2 0.2 0.2 0.2], 'Parent', ax9b);
    xlabel('Time (yr)'); ylabel('HCO_{3} (mol/L)'); grid on;
    xlim(log_xscale);
    pos = get(f9,'position');
    set(f9,'position',[pos(1) pos(2) pos(3)*1.5 pos(4)]);

    if PRINT_FIGURES
        saveas(f7,[subfolder filename ' 07 SO4 HCO3 Input' ],'png');
        saveas(f8,[subfolder filename ' 08 SO4 Concentrations' ],'png');
        saveas(f9,[subfolder filename ' 09 HCO3 Concentrations' ],'png');
    end
end

%initial oxygen concentration
O2_sat = 2.5e-4; %mol/L
%oxygen decrease limit
O2_999 = 0.999;
%uniform alpha distribution

```

```

O2_decay_5weeks = -1*log(1-O2_999)/(5/52.1429); % 5 weeks (in years)
O2_decay_5years = -1*log(1-O2_999)/(5); % 5 years
O2_decay = (unifrnd(O2_decay_5years,O2_decay_5weeks,[mc,1]));
%log uniform alpha distribution (not used in final code)
% Log_5weeks = log10(O2_decay_5weeks);
% Log_5years = log10(O2_decay_5years);
% O2_decay_log = 10.^(unifrnd(Log_5years,Log_5weeks,[mc,1]));

if INPUT_FIGURES
%   f10 = figure;
%   ax10a = axes('Parent',f10);
%   ax10b = axes('Parent',f10);
%   subplot(1,2,1,ax10a);
%   histogram(O2_decay_log, 'Parent', ax10a);
%   xlabel('Alpha (yr^{-1})'); ylabel('Frequency');
%   subplot(1,2,2,ax10b);
%   histogram(-(1./O2_decay_log).*log(.001), 'Parent', ax10b);
%   xlabel('Time to 99.9% Reduction (yr)'); ylabel('Frequency');
%   pos = get(f10,'position');
%   set(f10,'position',[pos(1) pos(2) pos(3)*1.5 pos(4)]);

f11 = figure;
ax11a = axes('Parent',f11);
ax11b = axes('Parent',f11);
subplot(1,2,1,ax11a);
histogram(O2_decay, 'Parent', ax11a);
xlabel('Alpha (yr^{-1})'); ylabel('Frequency');
subplot(1,2,2,ax11b);
histogram(-(1./O2_decay).*log(.001), 'Parent', ax11b);
xlabel('Time to 99.9% Reduction (yr)'); ylabel('Frequency');
pos = get(f11,'position');
set(f11,'position',[pos(1) pos(2) pos(3)*1.5 pos(4)]);

f13 = figure;
ax13a = axes('Parent',f13);
ax13b = axes('Parent',f13);
subplot(1,2,1,ax13a)
O2 = O2_sat .* exp(-(time_scale).*O2_decay);
plot(time_scale,O2,'color',[0.2 0.2 0.2 0.2], 'Parent', ax13a);
xlabel('Years'); ylabel('O_2 mol/L'); grid on;
linear_xscale = [0. 3];
xlim(linear_xscale);
subplot(1,2,2,ax13b)
semilogx(time_scale,O2,'color',[0.2 0.2 0.2 0.2], 'Parent', ax13b);
xlabel('Years'); ylabel('O_2 mol/L'); grid on;
O2_log_xscale = [1e-5 10];
xlim(O2_log_xscale);
pos = get(f13,'position');
set(f13,'position',[pos(1) pos(2) pos(3)*1.5 pos(4)]);

if PRINT_FIGURES
    %saveas(f10,[subfolder filename ' 10 Log Uniform Oxygen Decay' ],'png');
    saveas(f11,[subfolder filename ' 11 Uniform Oxygen Decay' ],'png');
    saveas(f13,[subfolder '13 Oxygen Concentration (uniform) ' filename],'png');
end
end

%only used if PotentialExp == 0 (Cong and Scully analysis)

```

```

%choose Eb and Erb constants based on normal distributions
Eb_const = normrnd( 1.11 ,0.061,[mc,1]);
Erp_const = normrnd(-0.0925,0.005,[mc,1]);

%choose pit growth expression A and n values
if PitGrowthExp == 0 %0 -- constant: 5e-6 m/year
    pgeA = ones(mc,1) * 5e-6;
    pgeN = ones(mc,1) * -10; %marker to distinguish pit growth method
elseif PitGrowthExp == 1 %Denison and Romanoff (1950) #65 Chino soil
    pgeA = normrnd( 89.2e-6,46.0e-6,[mc,1]);
    pgeN = normrnd( 0.57 , 0.26 ,[mc,1]);
    %ensure no negative numbers chosen
    pgeA(pgeA < 0) = 1e-7;
    pgeN(pgeN < 0) = 1e-7;
elseif PitGrowthExp == 2 %Denison and Romanoff (1950) #66 Mohave soil
    pgeA = normrnd(188.0e-6,41.0e-6,[mc,1]);
    pgeN = normrnd( 0.12 , 0.11 ,[mc,1]);
    %ensure no negative numbers chosen
    pgeA(pgeA < 0) = 1e-9;
    pgeN(pgeN < 0) = 1e-7;
end
toc;

```

A1.8 Overall Flow Control

In each iteration of the model, *mc*, run the code for determined number of years, *time*.

```

tic;
parfor p = 1:1:mc
%for p = 1:1:mc %for single threaded testing
    %use a sliced variable for parfor efficiency
    t = T(p,:);
    so4 = SO4(p,:);
    hco3 = HCO3(p,:);
    pitgrowth = zeros(1,length(time_scale)) - 20;
    pittime = zeros(1,length(time_scale));
    i = 1;
    while i <= length(time_scale)

        [pitgrowth(i),i_new] = ENV(trainedClassifier,i,t, pH(p,1),OH(p,1),CL(p),so4,
        ...
        hco3,O2_sat,O2_decay(p),Eb_const(p),Erp_const(p),time_scale,...
        pgeA(p),pgeN(p),PotentialExp,trainedModel_Eb, ...
        trainedModel_Erp, ClassifierMethod);

        pittime(i) = time_scale(i_new) - time_scale(i);

        if sum(pitgrowth(pitgrowth>0)) > CorrosionAllowance
            %exit the current simulation
            i_new = length(time_scale);
        end
        i = i_new + 1;
    end
    PitGrowth(p,:) = pitgrowth;
    PitTime(p,:) = pittime;
end
toc;

```


A1.9 Results

```
tic;
filename_time = strrep(datestr(datetime('now')),':','');
filename_ext = '.xlsx';
xl_filename = strcat(subfolder,filename,' output ',filename_time,filename_ext);
A = {'Constants','Value';'Realizations',mc;'KNN_num',KNN_num; ...
    'Total Time (yr)',time;'Num. Time Steps',length(time_scale);...
    'Pit-growth law',PitGrowthExp;'Potential Exp.',PotentialExp;...
    'RUN Number',RUN(6)};
xlRange = 'D1';
%output headers
xlswrite(xl_filename,A,1,xlRange);
A = {'Time Series (yr)',''};
xlRange = 'B1';
%output headers
xlswrite(xl_filename,A,1,xlRange);
xlRange = 'B2';
%output time_scale
xlswrite(xl_filename,time_scale',1,xlRange);

num_init = nnz(PitGrowth===-20);
num_active = nnz(PitGrowth===-15);
num_passive_Ecorr_lessthan_Eb = nnz(PitGrowth===-10);
num_passive_and_break_down = nnz(PitGrowth===-5);
num_pitgrowth = nnz(PitGrowth>0)
fraction_pitting = num_pitgrowth/mc

A = { 'Simulation Results for Generated Env Conditions',''; ...
    'Total Number of Active Conditions',num_active; ...
    'Total Number of Passive Conditions
(Ecorr<Eb)',num_passive_Ecorr_lessthan_Eb; ...
    'Total Number of Breakdown Events (but
Ecorr<Erp)',num_passive_and_break_down; ...
    'Total Number of Piting Initiation Events',num_pitgrowth; ...
    'Total Fraction Pitting Initiation',fraction_pitting; ...
    'Total Number of Piting events (after initiations)',num_init};
xlRange = 'D9';
%output active/passive data
xlswrite(xl_filename,A,1,xlRange);

NUM_INIT = zeros(mc,1);
NUM_ACTIVE = zeros(mc,1);
NUM_PASSIVE_ECORR_LESSTHAN_EB = zeros(mc,1);
NUM_PASSIVE_AND_BREAK_DOWN = zeros(mc,1);
NUM_PITGROWTH = zeros(mc,1);
R_PITDEPTH = zeros(mc,1);
for i = 1:mc
    NUM_INIT(i) = nnz(PitGrowth(i,:)==-20);
    NUM_ACTIVE(i) = nnz(PitGrowth(i,:)==-15);
    NUM_PASSIVE_ECORR_LESSTHAN_EB(i) = nnz(PitGrowth(i,:)==-10);
    NUM_PASSIVE_AND_BREAK_DOWN(i) = nnz(PitGrowth(i,:)==-5);
    NUM_PITGROWTH(i) = nnz(PitGrowth(i,*)>0);
```

```

    R_PITDEPTH(i) = sum(PitGrowth(i,PitGrowth(i,:)>0));
end
A = { 'Number of realizations with only active
condtions',nnz(NUM_ACTIVE==length(time_scale)); ...
      'Number of realizations with some active condtions',nnz(NUM_ACTIVE>0); ...
      'Number of realizations with no active condtions',nnz(NUM_ACTIVE==0); ...
      'Number of realizations with Passive conditions (but
Ecorr<Eb)',nnz(NUM_PASSIVE_ECORR_LESSTHAN_EB>0); ...
      'Number of realizations with Breakdown Events (but
Ecorr<Erp)',nnz(NUM_PASSIVE_AND_BREAK_DOWN>0); ...
      'Number of realizations with Pitting Events',nnz(NUM_PITGROWTH>0)};

xlRange = 'D17';
%output active/passive data
xlswrite(xl_filename,A,1,xlRange);
A = { 'Realization Number','Start Time (yr)','Duration (yr)','Depth of each pit
(m)',' ','Realization Number','Total Depth (m)'};
xlRange = 'D29';
xlswrite(xl_filename,A,1,xlRange);
%if there are pits, output time of pitting events
if ~isempty(PitTime(PitTime>0))
    [row,col,v] = find(PitTime);
    xlswrite(xl_filename,row,1,'D30');
    xlswrite(xl_filename,time_scale(col)',1,'E30');
    xlswrite(xl_filename,v,1,'F30');
end
%if there are pits, output pit depths
if ~isempty(PitGrowth(PitGrowth>0))
    xlswrite(xl_filename,PitGrowth(PitGrowth>0),1,'G30');
    [row1,col1,v1]= find(R_PITDEPTH);
    %PitDepth Stats
    threshold100mu = v1 <= 1e-4;
    numThreshold100mu = numel(threshold100mu(threshold100mu>0));
    threshold1mm = v1 >= 1e-3;
    numThreshold1mm = numel(threshold1mm(threshold1mm>0));

    A = { 'Number of realizations Pit Depth > 100 microns',numThreshold100mu; ...
          'Number of realizations Pit Depth > 1 mm',numThreshold1mm; ...
          'Max Pit Depth (m)', max(v1); ...
          'Mean Pit Depth (m)', mean(v1); ...
          'Pit Depth Standard Deviation (m)',std(v1)};

    xlswrite(xl_filename,A,1,'D23');
    xlswrite(xl_filename,row1,1,'I30');
    xlswrite(xl_filename,v1,1,'J30');
end

if AP_FIGURES

    if ClassifierMethod == 0
        IDX_KNN = zeros(mc,length(time_scale),KNN_num);
        parfor p = 1:1:mc
            idx_knn = zeros(length(time_scale),KNN_num);
            %variable slicing
            t = T(p,:);
            so4 = S04(p,:);
            hco3 = HCO3(p,:);
            for i = 1:1:length(time_scale)

```

```

        idx_knn(i,:) = knnsearch(Md2,[CL(p) so4(i) hco3(i) pH(p,1)
t(i)], 'k', KNN_num, 'Distance', 'seuclidean');
    end
    IDX_KNN(p, :, :) = idx_knn;
end
end

Active_Predict = zeros(mc, length(time_scale));
data_p_0 = cell(mc, 1);
data_p_1 = cell(mc, 1);

parfor p = 1:1:mc
    %variable slicing
    t = T(p, :);
    so4 = SO4(p, :);
    hco3 = HCO3(p, :);
    o2 = O2(p, :);
    oh = OH(p, :);
    apdata_predict = zeros(length(time_scale), 11);
    active_predict = zeros(length(time_scale), 1);
    switch ClassifierMethod
        case 0
            for i = 1:1:length(time_scale)
                active_predict(i) = predict(trainedClassifier, [CL(p) so4(i)
hco3(i) pH(p,1) t(i)]);
            end
            otherwise
                for i = 1:1:length(time_scale)
                    active_predict(i) =
trainedClassifier.predictFcn(table(CL(p), so4(i), hco3(i), pH(p,1), t(i), ...
'VariableNames', {'CL', 'SO4', 'HCO3', 'pH', 'T'}));
                end
            end

            apdata_predict(:, 1) = CL(p);
            apdata_predict(:, 2) = so4;
            apdata_predict(:, 3) = hco3;
            apdata_predict(:, 4) = pH(p,1);
            apdata_predict(:, 5) = t;
            apdata_predict(:, 6) = E_a_0 +
(2.3*R/F).*log10(((k_2_r*n_c*D_o2.*o2)./((k_a*n_a*D_cuc12).*(CL(p).^2))).*(t+273.15));
            apdata_predict(:, 7) = Eb_const(p) + (0.116 * log10(oh)) + (0.197 .*
log10(hco3)) - (0.130 .* log10(so4+CL(p)));
            apdata_predict(:, 8) = Erp_const(p) + 0.00373 * log10(oh) - 0.0139 .*
log10(hco3) - 0.0566 * log10(CL(p));
            apdata_predict(:, 9) = time_scale;
            apdata_predict(:, 10) =
trainedModel_Eb.predictFcn(table(apdata_predict(:, 1), so4, hco3, apdata_predict(:, 4), t
', ...
'VariableNames', {'CL', 'SO4', 'HCO3', 'pH', 'T'}));
            apdata_predict(:, 11) =
trainedModel_Erp.predictFcn(table(apdata_predict(:, 1), so4, hco3, apdata_predict(:, 4),
t', ...
'VariableNames', {'CL', 'SO4', 'HCO3', 'pH', 'T'}));

            data_p_0{p, 1} = apdata_predict(active_predict == 0, :);
            data_p_1{p, 1} = apdata_predict(active_predict == 1, :);
end
end

```

```

        Active_Predict(p,:) = active_predict;
    end

    data_0 = table2array(apdata(active == 0,:));
    data_1 = table2array(apdata(active == 1,:));

    %Temp(5) vs pH(4) vs Chloride(1) AP plot
    f24 = figure;
    ax24 = axes('Parent', f24);
    for i = 1:1:mc
        plot3(data_p_1{i,1}(:,5),data_p_1{i,1}(:,4),data_p_1{i,1}(:,1), ['b' '.'],
'Parent', ax24);
        hold on;
        plot3(data_p_0{i,1}(:,5),data_p_0{i,1}(:,4),data_p_0{i,1}(:,1), ['r' '.'],
'Parent', ax24);
    end
    end
    x1 = xlim; y1 = ylim; z1 = zlim;
    plot3(data_1(:,5),data_1(:,4),data_1(:,1), ['b' 'o'], 'Parent', ax24);
    plot3(data_0(:,5),data_0(:,4),data_0(:,1), ['r' 'o'], 'Parent', ax24);
    hold off;
    xlim(x1); ylim(y1); zlim(z1); axis square; grid on;
    xlabel('Temperature (\circC)'); ylabel('pH'); zlabel('Cl^{-} (mol/L)');
    f25 = copyobj(gcf,0); view([45 30]);

    %Temp(5) vs pH(4) vs SO4(2) AP plot
    f26 = figure;
    ax26 = axes('Parent', f26);
    for i = 1:1:mc
        plot3(data_p_1{i,1}(:,5),data_p_1{i,1}(:,4),data_p_1{i,1}(:,2), ['b' '.'],
'Parent', ax26);
        hold on;
        plot3(data_p_0{i,1}(:,5),data_p_0{i,1}(:,4),data_p_0{i,1}(:,2), ['r' '.'],
'Parent', ax26);
    end
    end
    x1 = xlim; y1 = ylim; z1 = zlim;
    plot3(data_1(:,5),data_1(:,4),data_1(:,2), ['b' 'o'], 'Parent', ax26);
    plot3(data_0(:,5),data_0(:,4),data_0(:,2), ['r' 'o'], 'Parent', ax26);
    hold off;
    xlim(x1); ylim(y1); zlim(z1); axis square; grid on;
    xlabel('Temperature (\circC)'); ylabel('pH'); zlabel('SO_4 (mol/L)');
    f27 = copyobj(gcf,0); view([45 30]);

    %Temp(5) vs pH(4) vs HCO3(3) AP plot
    f28 = figure;
    ax28 = axes('Parent', f28);
    for i = 1:1:mc
        %figure(R1);
        plot3(data_p_1{i,1}(:,5),data_p_1{i,1}(:,4),data_p_1{i,1}(:,3), ['b' '.'],
'Parent', ax28);
        hold on;
        plot3(data_p_0{i,1}(:,5),data_p_0{i,1}(:,4),data_p_0{i,1}(:,3), ['r' '.'],
'Parent', ax28);
    end
    end
    x1 = xlim; y1 = ylim; z1 = zlim;
    plot3(data_1(:,5),data_1(:,4),data_1(:,3), ['b' 'o'], 'Parent', ax28);
    plot3(data_0(:,5),data_0(:,4),data_0(:,3), ['r' 'o'], 'Parent', ax28);
    hold off;

```

```

xlim(x1); ylim(y1); zlim(z1); axis square; grid on;
xlabel('Temperature (\circ C)'); ylabel('pH'); zlabel('HCO_3 (mol/L)');
f29 = copyobj(gcf,0); view([45 30]);

%Time(9) vs Voltage AP plot
f30 = figure;
ax30 = axes('Parent', f30);
for i = 1:1:mc
    semilogx(data_p_1{i,1}(:,9),data_p_1{i,1}(:,6), ['b' '.'], 'Parent', ax30);
    hold on;
    semilogx(data_p_0{i,1}(:,9),data_p_0{i,1}(:,6), ['r' 's'], 'Parent', ax30);
end
%dummy plot to correct legend
h = zeros(2, 1);
h(1) = semilogx(0,0,['b' '.']);
h(2) = semilogx(0,0,['r' 's']);
hold off;
xlim([1e-3 time]); ylim([-0.5 0.2]); grid on;
xlabel('Time (yr^{-1})'); ylabel('E_{corr} (V_{SCE})');
legend([h(1),h(2)], 'E_{corr} Active', 'E_{corr} Passive', ...
        'Location','eastoutside');
pos = get(gcf,'position');
set(gcf,'position',[pos(1:2)/1.5 pos(3)*1.5 pos(4)*1]);

%Time(9) vs Voltage AP plot
f31 = figure;
ax31 = axes('Parent', f31);
for i = 1:1:mc
    %semilogx(data_p_1{i,1}(:,9),data_p_1{i,1}(:,7), ['b' '+'], 'Parent', ax31);
    %hold on;
    %semilogx(data_p_1{i,1}(:,9),data_p_1{i,1}(:,8), ['b' 'x'], 'Parent', ax31);
    semilogx(data_p_0{i,1}(:,9),data_p_0{i,1}(:,7), ['k' '.'], 'Parent', ax31);
    hold on;
    semilogx(data_p_0{i,1}(:,9),data_p_0{i,1}(:,8), ['r' 'o'], 'Parent', ax31);
end
%dummy plot to correct legend
h = zeros(2, 1);
%h(1) = semilogx(0,0,['b' '+']);
%h(2) = semilogx(0,0,['b' 'x']);
h(1) = semilogx(0,0,['k' '.']);
h(2) = semilogx(0,0,['r' 'o']);
hold off;
xlim([1e-3 time]); grid on; set(gca,'XScale','log');
xlabel('Time (yr^{-1})'); ylabel('Cong and Scully Eq. (V_{SCE})');
legend([h(1),h(2)], ...
        'E_b Passive', 'E_{rp} Passive', 'Location','eastoutside');
%legend([h(1),h(2),h(3),h(4)], 'E_b Active', 'E_{rp} Active', ...
%      'E_b Passive', 'E_{rp} Passive', 'Location','eastoutside');
pos = get(gcf,'position');
set(gcf,'position',[pos(1:2)/1.5 pos(3)*1.5 pos(4)*1]);

%Time(9) vs Voltage AP plot
f32 = figure;
ax32 = axes('Parent', f32);
for i = 1:1:mc
    %semilogx(data_p_1{i,1}(:,9),data_p_1{i,1}(:,7), ['b' '+'], 'Parent', ax32);
    %hold on;
    %semilogx(data_p_1{i,1}(:,9),data_p_1{i,1}(:,8), ['b' 'x'], 'Parent', ax32);

```

```

        semilogx(data_p_1{i,1}(:,9),data_p_1{i,1}(:,6), ['b' '.'], 'Parent', ax32);
        hold on;
        semilogx(data_p_0{i,1}(:,9),data_p_0{i,1}(:,7), ['k' '.'], 'Parent', ax32);
        semilogx(data_p_0{i,1}(:,9),data_p_0{i,1}(:,8), ['r' 'o'], 'Parent', ax32);
        semilogx(data_p_0{i,1}(:,9),data_p_0{i,1}(:,6), ['r' 's'], 'Parent', ax32);
    end
    %dummy plot to correct legend
    h = zeros(4, 1);
    %h(1) = semilogx(0,0,['b' '+']);
    %h(2) = semilogx(0,0,['b' 'x']);
    h(1) = semilogx(0,0,['b' '.']);
    h(2) = semilogx(0,0,['k' '.']);
    h(3) = semilogx(0,0,['r' 'o']);
    h(4) = semilogx(0,0,['r' 's']);
    hold off;
    xlim([1e-3 time]); ylim([-0.5 0.2]); grid on;
    xlabel('Time (yr^{-1})'); ylabel('Cong and Scully Eq. (V_{SCE})');
    %legend([h(1),h(2),h(3),h(4),h(5),h(6)], 'E_b Active', 'E_{rp} Active', ...
    %      'E_{corr} Active', 'E_b Passive', 'E_{rp} Passive', 'E_{corr}
Passive', 'Location', 'eastoutside');
    legend([h(1),h(2),h(3),h(4)], ...
           'E_{corr} Active', 'E_b Passive', 'E_{rp} Passive', 'E_{corr}
Passive', 'Location', 'eastoutside');
    pos = get(gcf, 'position');
    set(gcf, 'position', [pos(1:2)/1.5 pos(3)*1.5 pos(4)*1]);

    %Time(9) vs Temperature(5) AP plot
    f33 = figure;
    ax33 = axes('Parent', f33);
    for i = 1:1:mc
        semilogx(data_p_1{i,1}(:,9),data_p_1{i,1}(:,5), ['b' '.'], 'Parent', ax33);
        hold on;
        semilogx(data_p_0{i,1}(:,9),data_p_0{i,1}(:,5), ['r' 'o'], 'Parent', ax33);
    end
    %dummy plot to correct legend
    h = zeros(2, 1);
    h(1) = semilogx(0,0,['b' '.']);
    h(2) = semilogx(0,0,['r' 'o']);
    hold off;
    if time <= 1e-2 xmin = time/100; else xmin = 1e-2; end
    xlim([xmin time]); grid on;
    xlabel('Time (yr^{-1})'); ylabel('T (\circ C)');
    legend([h(1),h(2)], 'Active', 'Passive', 'Location', 'eastoutside');
    pos = get(gcf, 'position');
    set(gcf, 'position', [pos(1:2)/1.5 pos(3)*1.5 pos(4)*1]);

    %2D matrix of plots for all components of apdata
    f34 = figure;
    varNames = {'Cl^{-}'; 'SO_4'; 'HCO_3'; 'pH'; 'Temperature'};
    gplotmatrix(table2array(apdata), [], active, ['r' 'b'], 'o.', [4
8], false, [], varNames, varNames);
    title('UWO Data: Active - Closed Blue Circles; Passive - Open Red Circles');
    pos = get(gcf, 'position');
    set(gcf, 'position', [pos(1:2)/3 pos(3)*2 pos(4)*2]);

    %2D matrix of plots for all components of generated data and predicted
    f35 = figure;
    APdata_predict = zeros(length(time_scale)*mc,6);

```

```

Active_Predict_Vector = zeros(length(time_scale)*mc,1);
for i = 1:mc
    APdata_predict((i-1)*length(time_scale)+1:i*length(time_scale),1) = CL(i);
    APdata_predict((i-1)*length(time_scale)+1:i*length(time_scale),2) = SO4(i,:);
    APdata_predict((i-1)*length(time_scale)+1:i*length(time_scale),3) =
HCO3(i,:);
    APdata_predict((i-1)*length(time_scale)+1:i*length(time_scale),4) = pH(i);
    APdata_predict((i-1)*length(time_scale)+1:i*length(time_scale),5) = T(i,:);
    APdata_predict((i-1)*length(time_scale)+1:i*length(time_scale),6) =
time_scale;
    Active_Predict_Vector((i-1)*length(time_scale)+1:i*length(time_scale),1) =
Active_Predict(i,:);
end
varNames = {'Cl^-'; 'SO_4'; 'HCO_3'; 'pH'; 'Temperature'; 'Time'};
gplotmatrix(APdata_predict,[],Active_Predict_Vector,['r' 'b'],'o.',[4
6],false,[],varNames,varNames);
title('Generated Data: Active - Closed Blue Circles; Passive - Open Red
Circles');
pos = get(gcf,'position');
set(gcf,'position',[pos(1:2)/7 pos(3)*2.4 pos(4)*2.4]);

%Time(9) vs pH(4) vs Chloride(1) AP plot
f36 = figure;
ax36 = axes('Parent', f36);
for i = 1:1:mc
    plot3(data_p_1{i,1}(:,9),data_p_1{i,1}(:,4),data_p_1{i,1}(:,1), ['b' '.'],
'Parent', ax36);
    hold on;
    plot3(data_p_0{i,1}(:,9),data_p_0{i,1}(:,4),data_p_0{i,1}(:,1), ['r' '.'],
'Parent', ax36);
end
x1 = xlim; y1 = ylim; z1 = zlim;
%plot3(data_1(:,9),data_1(:,4),data_1(:,1), ['b' 'o'], 'Parent', ax36);
%plot3(data_0(:,9),data_0(:,4),data_0(:,1), ['r' 'o'], 'Parent', ax36);
hold off;
xlim(x1); ylim(y1); zlim(z1); axis square; grid on;
xlabel('Time (yr)'); ylabel('pH'); zlabel('Cl^- (mol/L)')
f37 = copyobj(gcf,0); view([45 30]);
f38 = copyobj(gcf,0); view([135 30]);

%Active/Passive transitions with time
set(groot,'defaultAxesLineStyleOrder',{'-','--',':','-.'});
f39 = figure;
ax39 = axes('Parent',f39);
adjPitGrowth = PitGrowth;
adjPitGrowth(adjPitGrowth == -20) = 0;
semilogx(time_scale,adjPitGrowth,'Parent',ax39);
if time <= 1e-2 xmin = time/100; else xmin = 1e-2; end
xlim([xmin time]); ylim([-16 1]); grid on;
yticks([-15 -10 -5 0]);
yticklabels({'Active','Passive (E_{corr}<E_b)','Passive (E_{corr}>E_b and
E_{corr}<E_{rp})','Pitting (E_{corr}>E_{rp})'});
xlabel('Time (yr)');
set(gcf,'position',[pos(1)/4 pos(2) pos(3)*1.5 pos(4)]);
set(groot,'defaultAxesLineStyleOrder','remove')

%Time(9) vs UWO Voltage AP plot
f40 = figure;

```

```

ax40 = axes('Parent', f40);
for i = 1:1:mc
    %semilogx(data_p_1{i,1}(:,9),data_p_1{i,1}(:,10), ['b' '+'], 'Parent', ax40);
    %hold on;
    %semilogx(data_p_1{i,1}(:,9),data_p_1{i,1}(:,11), ['b' 'x'], 'Parent', ax40);
    semilogx(data_p_1{i,1}(:,9),data_p_1{i,1}(:,6), ['b' '.'], 'Parent', ax40);
    hold on;
    semilogx(data_p_0{i,1}(:,9),data_p_0{i,1}(:,10), ['k' '.'], 'Parent', ax40);
    semilogx(data_p_0{i,1}(:,9),data_p_0{i,1}(:,11), ['r' 'o'], 'Parent', ax40);
    semilogx(data_p_0{i,1}(:,9),data_p_0{i,1}(:,6), ['r' 's'], 'Parent', ax40);
end
%dummy plot to correct legend
h = zeros(4, 1);
%h(1) = semilogx(0,0,['b' '+']);
%h(2) = semilogx(0,0,['b' 'x']);
h(1) = semilogx(0,0,['b' '.']);
h(2) = semilogx(0,0,['k' '.']);
h(3) = semilogx(0,0,['r' 'o']);
h(4) = semilogx(0,0,['r' 's']);
hold off;
xlim([1e-3 time]); ylim([-0.5 0.2]); grid on;
xlabel('Time (yr^{-1})'); ylabel('UWO Trained Data (V_{SCE})');
% legend([h(1),h(2),h(3),h(4),h(5),h(6)], 'E_b Active', 'E_{rp} Active', ...
% 'E_{corr} Active', 'E_b Passive', 'E_{rp} Passive', 'E_{corr}
Passive', 'Location', 'eastoutside');
legend([h(1),h(2),h(3),h(4)], ...
'E_{corr} Active', 'E_b Passive', 'E_{rp} Passive', 'E_{corr}
Passive', 'Location', 'eastoutside');
pos = get(gcf, 'position');
set(gcf, 'position', [pos(1:2)/1.5 pos(3)*1.5 pos(4)*1]);

%Time(9) vs UWO voltage AP plot
f41 = figure;
ax41 = axes('Parent', f41);
for i = 1:1:mc
    %semilogx(data_p_1{i,1}(:,9),data_p_1{i,1}(:,7), ['b' '+'], 'Parent', ax41);
    %hold on;
    %semilogx(data_p_1{i,1}(:,9),data_p_1{i,1}(:,8), ['b' 'x'], 'Parent', ax41);
    semilogx(data_p_0{i,1}(:,9),data_p_0{i,1}(:,10), ['k' '.'], 'Parent', ax41);
    hold on;
    semilogx(data_p_0{i,1}(:,9),data_p_0{i,1}(:,11), ['r' 'o'], 'Parent', ax41);
end
%dummy plot to correct legend
h = zeros(2, 1);
%h(1) = semilogx(0,0,['b' '+']);
%h(2) = semilogx(0,0,['b' 'x']);
h(1) = semilogx(0,0,['k' '.']);
h(2) = semilogx(0,0,['r' 'o']);
hold off;
xlim([1e-3 time]); grid on; set(gca, 'XScale', 'log');
xlabel('Time (yr^{-1})'); ylabel('UWO Trained Data (V_{SCE})');
legend([h(1),h(2)], ...
'E_b Passive', 'E_{rp} Passive', 'Location', 'eastoutside');
%legend([h(1),h(2),h(3),h(4)], 'E_b Active', 'E_{rp} Active', ...
% 'E_b Passive', 'E_{rp} Passive', 'Location', 'eastoutside');
pos = get(gcf, 'position');
set(gcf, 'position', [pos(1:2)/1.5 pos(3)*1.5 pos(4)*1]);

```



```

if ClassifierMethod == 0
    %percent active
    mean_active_neighbours = mean(mean(sum((active(IDX_KNN(:, :, :)) == 1, 3)));
    mean_passive_neighbours = mean(mean(sum((active(IDX_KNN(:, :, :)) == 0, 3)));
    A = {'Mean Active Neighbours', mean_active_neighbours; 'Mean Passive
Neighbours', mean_passive_neighbours};
    xlRange = 'M23';
    xlswrite(xl_filename, A, 1, xlRange);
end

dsa = APData(:, {'Cl', 'SO4', 'HCO3', 'pH', 'T', 'active'});
statarray_1 = grpstats(dsa, 'active');
statarray_2 = grpstats(dsa, {'active', 'T'}, {'min', 'max'}, 'DataVars', 'pH');
statarray_3 = grpstats(dsa, {'active', 'T'}, {'min', 'max'}, 'DataVars', 'Cl');
statarray_4 = grpstats(dsa, {'active', 'T'}, {'min', 'max'}, 'DataVars', 'SO4');
statarray_5 = grpstats(dsa, {'active', 'T'}, {'min', 'max'}, 'DataVars', 'HCO3');

xlRange = 'M26';
writetable(statarray_1, xl_filename, 'Sheet', 1, 'Range', xlRange);
xlRange = 'M30';
writetable(statarray_2, xl_filename, 'Sheet', 1, 'Range', xlRange);
xlRange = 'M42';
writetable(statarray_3, xl_filename, 'Sheet', 1, 'Range', xlRange);
xlRange = 'M55';
writetable(statarray_4, xl_filename, 'Sheet', 1, 'Range', xlRange);
xlRange = 'M66';
writetable(statarray_5, xl_filename, 'Sheet', 1, 'Range', xlRange);

if PRINT_FIGURES
    saveas(f24, [subfolder filename ' 24 Temp vs pH vs Cl AP Plot'], 'png');
    saveas(f25, [subfolder filename ' 25 Temp vs pH vs Cl AP Plot View 2'
], 'png');
    saveas(f26, [subfolder filename ' 26 Temp vs pH vs SO4 AP Plot' ], 'png');
    saveas(f27, [subfolder filename ' 27 Temp vs pH vs SO4 AP Plot View 2'
], 'png');
    saveas(f28, [subfolder filename ' 28 Temp vs pH vs HCO3 AP Plot' ], 'png');
    saveas(f29, [subfolder filename ' 29 Temp vs pH vs HCO3 AP Plot View 2'
], 'png');
    saveas(f30, [subfolder filename ' 30 Time vs Ecorr AP Plot CongScully'
], 'png');
    saveas(f31, [subfolder filename ' 31 Time vs Eb and Erp AP Plot CongScully'
], 'png');
    saveas(f32, [subfolder filename ' 32 Time vs Eb Erp and Ecorr AP Plot
CongScully' ], 'png');
    saveas(f33, [subfolder filename ' 33 Time vs Temperature AP Plot' ], 'png');
    saveas(f34, [subfolder filename ' 34 UWO Data Matrix Plot' ], 'png');
    saveas(f35, [subfolder filename ' 35 Generated Data Matrix Plot' ], 'png');
    saveas(f36, [subfolder filename ' 36 Time vs pH vs Cl AP Plot' ], 'png');
    saveas(f37, [subfolder filename ' 37 Time vs pH vs Cl AP Plot View 2'
], 'png');
    saveas(f38, [subfolder filename ' 38 Time vs pH vs Cl AP Plot View 3'
], 'png');
    saveas(f39, [subfolder filename ' 39 Active_Passive over time Plot' ], 'png');
    saveas(f40, [subfolder filename ' 40 Time vs Eb Erp and Ecorr AP Plot UWO'
], 'png');
    saveas(f41, [subfolder filename ' 41 Time vs Eb and Erp AP Plot UWO' ], 'png');
end
end

```

```

f42 = figure;
ax42 = axes('Parent',f42);
adjPitGrowth = PitGrowth;
adjPitGrowth(adjPitGrowth ==-20) = 0;
numel_active(:) = sum(adjPitGrowth(:, :) == -15);
numel_passive(:) = sum(adjPitGrowth(:, :) == -10);
numel_pitting(:) = sum(adjPitGrowth(:, :) > -5);
semilogx(time_scale,numel_active,'b','Linewidth',2,'Parent',ax42);
hold on;
semilogx(time_scale,numel_passive,'k','Linewidth',2,'Parent',ax42);
semilogx(time_scale,numel_pitting,'r','Linewidth',2,'Parent',ax42);
hold off;
if time <= 1e-2 xmin = time/100; else xmin = 1e-2; end
xlim([xmin time]); grid on;
xlabel('Time (yr)'); ylabel('Realizations');
legend('Number of Active Realizations', 'Number of Passive Realizations', 'Number of
Pitting Realizations', 'Location', 'best');
pos = get(gcf, 'position');
set(gcf, 'position', [pos(1)/4 pos(2) pos(3)*1.5 pos(4)]);

f43 = figure;
ax43 = axes('Parent',f43);
bins = logspace(min(floor(log10(v1))),max(ceil(log10(v1))),50);
histogram(ax43,v1,bins);
grid on;
xlabel('Pit Depth (m)'); ylabel('Frequency');
set(gca, 'XMinorTick', 'on', 'XScale', 'log');

f44 = figure;
ax44 = axes('Parent',f44);
histogram(ax44,v1);
xlabel('Pit Depth (m)'); ylabel('Frequency');

saveas(f42,[subfolder filename ' 42 Active_Passive SUM over time Plot' ],'png');
saveas(f43,[subfolder filename ' 43 Pit Depth Histogram Semilog' ],'png');
saveas(f44,[subfolder filename ' 44 Pit Depth Histogram' ],'png');

toc;

%end associated function handle
%end

```

A1.10 Environment Function

Given a set of parameters, determine if the system is active or passive.

```

function [pitgrowth, i_new] = ENV(trainedclassifier,i,t,ph,oh,cl,so4,hco3,o2_sat, ...
                                o2_decay,Eb_const,Erp_const, ...
                                time_scale,pgeA,pgeN,potentialexp,...
                                trainedModel_Eb,trainedModel_Erp, ...
                                classifiermethod)

%predict: 0 passive; 1 active
switch classifiermethod
case 0
    if (predict(trainedclassifier,[cl so4(i) hco3(i) ph t(i)])==0)
        [pitgrowth, i_new] = PIT(i,t,ph,oh,cl,so4,hco3,o2_sat, ...

```

```

                                O2_decay,Eb_const,Erp_const, ...
                                time_scale,pgeA,pgeN,potentialexp, ...
                                trainedModel_Eb,trainedModel_Erp);

else %if active
    pitgrowth = -15;
    i_new = i;
end
otherwise
if (trainedClassifier.predictFcn(table(c1,so4(i),hco3(i),ph,t(i), ...
'variableNames',{'c1','so4','hco3','ph','t'}))==0)
    [pitgrowth, i_new] = PIT(i,t,ph,oh,c1,so4,hco3,O2_sat, ...
                                O2_decay,Eb_const,Erp_const, ...
                                time_scale,pgeA,pgeN,potentialexp, ...
                                trainedModel_Eb,trainedModel_Erp);

else %if active
    pitgrowth = -15;
    i_new = i;
end
end
end
end

```

A1.11 Pit Initiation Function

Given a set of parameters determine if a pit is initialized.

The breakdown potential is determined by: (from Cong et al. 2009: Equation 8)
 $E_B = 1.11 + 0.116 \times \log[\text{OH}^-] + 0.197 \times \log[\text{HCO}_3^-] - 0.130 \times \log([\text{SO}_4^{2-}] + [\text{Cl}^-])$
or by regression of UWO data.

```

function [pitgrowth, i_new] = PIT(i,t,ph,oh,c1,so4,hco3,O2_sat, ...
                                O2_decay,Eb_const,Erp_const, ...
                                time_scale,pgeA,pgeN,potentialexp,...
                                trainedModel_Eb,trainedModel_Erp)

%Ecorr Calculation constants Eq 14 from King et al. 1995
F      = 96487;    %C/mol Faraday constant
R      = 8.314;    %J/K/mol Gas Constant
n_a    = 1;       %number of anodic electrons
n_c    = 4;       %number of cathodic electrons
E_a_0  = -0.105;  %V(SCE) standard potential
k_a    = 3.3e-4;  %dm4/mol/s
k_2_r  = 1.42e-3; %dm/s
D_O2   = 1.7e-5; %cm2/s
D_CuCl2 = 5.5e-6; %cm2/s

O2 = O2_sat * exp(-1 * time_scale(i) * O2_decay);
Ecorr = E_a_0 + 2.3*R*(t(i)+273.15)* ...
        log10((k_2_r*n_c*D_O2*O2)/(k_a*c1^2*n_a*D_CuCl2))/F;

if potentialexp == 0
    Eb = Eb_const+0.116*log10(oh)+0.197*log10(hco3(i))-0.130*log10(so4(i)+c1);
elseif potentialexp == 1
    Eb = trainedModel_Eb.predictFcn(table(c1,so4(i),hco3(i),ph,t(i), ...
'variableNames',{'c1','so4','hco3','ph','t'}));
end
end

```

```

if Ecorr>Eb
    [pitgrowth, i_new] = PITPROP(i,t,ph,oh,c1,so4,hco3,O2_sat, ...
                                O2_decay,Erp_const,Ecorr, ...
                                time_scale,pgeA,pgeN,potentialexp,...
                                trainedModel_Erp);
    %[pitgrowth, i_new] = PITPROP(i,t,oh,so4,hco3,O2_sat, ...
    %                                O2_decay,Erp_const,Ecorr,time_scale);
else
    pitgrowth = -10;
    i_new = i;
end
end

```

A1.12 Pit Propagation Function

Given a set of parameters determine if a pit propagates.

The repassivation potential is determined by: (from Cong et al. 2009: Equation 5)

$$E_{RP} = -0.0925 + 0.00373 \times \log[\text{OH}^-] - 0.0139 \times \log[\text{HCO}_3^-] - 0.0566 \times \log[\text{Cl}^-]$$

or by regression of UWO data.

```

function [pitgrowth, i_new] = PITPROP(i,t,ph,oh,c1,so4,hco3,O2_sat, ...
                                O2_decay,Erp_const,EcorrPit, ...
                                time_scale,pgeA,pgeN,potentialexp,...
                                trainedModel_Erp)

%Ecorr Calculation constants Eq 14 from King et al. 1995
F      = 96487; %C/mol Faraday constant
R      = 8.314; %J/K/mol Gas Constant
n_a    = 1; %number of anodic electrons
n_c    = 4; %number of cathodic electrons
E_a_0  = -0.105; %V(SCE) standard potential
k_a    = 3.3e-4; %dm4/mol/s
k_2_r  = 1.42e-3; %dm/s
D_O2   = 1.7e-5; %cm2/s
D_CuCl2 = 5.5e-6; %cm2/s

%calculate re-passivation potential
if potentialexp == 0
    Erp = Erp_const + 0.00373 * log10(oh) - 0.0139 ...
          * log10(hco3(i)) - 0.0566 * log10(c1);
elseif potentialexp == 1
    Erp = trainedModel_Erp.predictFcn(table(c1,so4(i),hco3(i),ph,t(i), ...
          'VariableNames',{'Cl','SO4','HCO3','pH','T'}));
end

if (EcorrPit>Erp) && (i<length(time_scale))
    %Pit Growth Expression Method
    if pgeN == -10 % constant growth method
        if i == 1
            pitgrowth = pgeA*(time_scale(i)-0);
        else
            pitgrowth = pgeA*(time_scale(i)-time_scale(i-1));
        end
    else % Denison and Romanoff (1950) method
        if i == 1
            pitgrowth = pgeA*(time_scale(i)-0)^pgeN;

```

```

else
    pitgrowth = pgeA*(time_scale(i)-time_scale(i-1))^pgeN;
end
end
%pitgrowth = pitgrowthrate;

i_new = i+1; %increment pit growth time

%Calculate new EcorrGrowth
O2 = O2_sat * exp(-1 * time_scale(i_new) * O2_decay);
EcorrGrowth = E_a_0 + 2.3*R*(t(i_new)+273.15)* ...
    log10((k_2_r*n_c*D_O2*O2)/(k_a*c_l^2*n_a*D_CuCl2))/F;

%Calculate new re-passivation potential
if potentialexp == 0
    Erp = Erp_const + 0.00373 * log10(oh) - 0.0139 * ...
        log10(hco3(i_new)) - 0.0566 * log10(c_l);
elseif potentialexp == 1
    Erp = trainedModel_Erp.predictFcn(table(c_l,so4(i),hco3(i),ph,t(i_new),
...
        'VariableNames',{'c_l','so4','hco3','ph','T'}));
end
while EcorrGrowth>Erp
    i_new = i_new + 1; %increment pit growth time
    if i_new >= length(time_scale)
        i_new = length(time_scale);
        break %if at the end of time
    end
    %re-calculate parameters for new time
    O2 = O2_sat * exp(-1 * time_scale(i_new) * O2_decay);
    EcorrGrowth = E_a_0 + 2.3*R*(t(i_new)+273.15)* ...
        log10((k_2_r*n_c*D_O2*O2)/(k_a*c_l^2*n_a*D_CuCl2))/F;
    if potentialexp == 0
        Erp = Erp_const + 0.00373 * log10(oh) - 0.0139 * ...
            log10(hco3(i_new)) - 0.0566 * log10(c_l);
    elseif potentialexp == 1
        Erp =
trainedModel_Erp.predictFcn(table(c_l,so4(i),hco3(i),ph,t(i_new), ...
            'VariableNames',{'c_l','so4','hco3','ph','T'}));
    end
end
if pgeN == -10 % constant growth method
    pitgrowth = pitgrowth + pgeA*(time_scale(i_new)-time_scale(i));
else % Denison and Romanoff (1950) method
    pitgrowth = pitgrowth + pgeA*(time_scale(i_new)-time_scale(i))^pgeN;
end
%pitgrowth = pitgrowth + 1;
else
    pitgrowth = -5;
    i_new = i;
end
end
end

```

A2 importfile function

```
function tableout = importfile(workbookFile, sheetName, startRow, endRow)

%IMPORTFILE Import data from a spreadsheet
% DATA = IMPORTFILE(FILE) reads data from the first worksheet in the
% Microsoft Excel spreadsheet file named FILE and returns the data as a
% table.
%
% DATA = IMPORTFILE(FILE,SHEET) reads from the specified worksheet.
%
% DATA = IMPORTFILE(FILE,SHEET,STARTROW,ENDROW) reads from the specified
% worksheet for the specified row interval(s). Specify STARTROW and
% ENDROW as a pair of scalars or vectors of matching size for
% dis-contiguous row intervals. To read to the end of the file specify an
% ENDROW of inf.
%
%           Non-numeric cells are replaced with: NaN
%
% Example:
%   ActivePassiveDataTest = importfile('ActivePassiveDataTest.xlsx','Sheet1',5,321);
%
%   See also XLSREAD.

% Auto-generated by MATLAB on 2018/05/07 13:22:33
```

A2.1 Input handling

```
% If no sheet is specified, read first sheet
if nargin == 1 || isempty(sheetName)
    sheetName = 1;
end

% If row start and end points are not specified, define defaults
if nargin <= 3
    startRow = 5;
    endRow = 628;
end
```

A2.2 Import the data, extracting spreadsheet dates in Excel serial date format

```
[~, ~, raw, dates] = xlsread(workbookFile, sheetName,
    sprintf('A%d:Q%d', startRow(1), endRow(1)), ' ', @convertSpreadsheetExcelDates);
for block=2:length(startRow)
    [~, ~, tmpRawBlock, tmpDateNumBlock] = xlsread(workbookFile, sheetName,
    sprintf('A%d:Q%d', startRow(block), endRow(block)), ' ', @convertSpreadsheetExcelDates);
    raw = [raw; tmpRawBlock]; %#ok<AGROW>
    dates = [dates; tmpDateNumBlock]; %#ok<AGROW>
end
raw(cellfun(@(x) ~isempty(x) && isnumeric(x) && isnan(x), raw)) = {' '};
stringVectors = string(raw(:, [2, 12]));
stringVectors(ismissing(stringVectors)) = ' ';
raw = raw(:, [3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17]);
dates = dates(:, 1);
```

A2.3 Replace non-numeric cells with NaN

```
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),raw); % Find non-numeric cells
raw(R) = {NaN}; % Replace non-numeric cells
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),dates); % Find non-numeric cells
dates(R) = {NaN}; % Replace non-numeric Excel dates with NaN
```

A2.4 Create output variable

```
I = cellfun(@(x) ischar(x), raw);
raw(I) = {NaN};
data = reshape([raw{:}],size(raw));
```

A2.5 Create table

```
tableout = table;
```

A2.6 Allocate imported array to column variable names

```
dates(~cellfun(@(x) isnumeric(x) || islogical(x), dates)) = {NaN};
tableout.date = datetime([dates{:},1]','ConvertFrom','Excel');
tableout.FileName = stringVectors(:,1);
tableout.Cl = data(:,1);
tableout.SO4 = data(:,2);
tableout.HCO3 = data(:,3);
tableout.pH = data(:,4);
tableout.T = data(:,5);
tableout.Ecorr = data(:,6);
tableout.E1 = data(:,7);
tableout.Eb = data(:,8);
tableout.Erp = data(:,9);
tableout.Note = categorical(stringVectors(:,2));
tableout.active = data(:,10);
tableout.I01 = data(:,11);
tableout.m1 = data(:,12);
tableout.I02 = data(:,13);
tableout.m2 = data(:,14);

% For code requiring serial dates (datetime) instead of datetime, uncomment
% the following line(s) below to return the imported dates as datetime(s).

% tableout.date=datetime(tableout.date);
```

A3 importChlorideCDF function

```
function tableout = importChlorideCDF(workbookFile, sheetName, startRow, endRow)
```

```
%IMPORTFILE Import data from a spreadsheet
% DATA = IMPORTFILE(FILE) reads data from the first worksheet in the
% Microsoft Excel spreadsheet file named FILE and returns the data as a
% table.
%
% DATA = IMPORTFILE(FILE,SHEET) reads from the specified worksheet.
%
% DATA = IMPORTFILE(FILE,SHEET,STARTROW,ENDROW) reads from the specified
% worksheet for the specified row interval(s). Specify STARTROW and
% ENDROW as a pair of scalars or vectors of matching size for
% dis-contiguous row intervals. To read to the end of the file specify an
% ENDROW of inf.%
% Example:
% ChlorideCDF = importfile('ChlorideCDF.xlsx','Sheet1',2,44);
%
% See also XLSREAD.

% Auto-generated by MATLAB on 2018/05/29 11:01:38
```

A3.1 Input handling

```
% If no sheet is specified, read first sheet
if nargin == 1 || isempty(sheetName)
    sheetName = 1;
end

% If row start and end points are not specified, define defaults
if nargin <= 3
    startRow = 2;
    endRow = 44;
end
```

A3.2 Import the data

```
data = xlsread(workbookFile, sheetName, sprintf('A%d:D%d',startRow(1),endRow(1)));
for block=2:length(startRow)
    tmpDataBlock = xlsread(workbookFile, sheetName,
    sprintf('A%d:D%d',startRow(block),endRow(block)));
    data = [data;tmpDataBlock]; %#ok<AGROW>
end
```

A3.3 Create table

```
tableout = table;
```


A3.4 Allocate imported array to column variable names

```
tableout.Clmo11 = data(:,1);
tableout.SortCl = data(:,2);
tableout.PointNo = data(:,3);
tableout.Partofdistr = data(:,4);
```

A4 trainClassifier_Bagged_Tree_Ver function

```
function [trainedClassifier, validationAccuracy] =
trainClassifier_Bagged_Tree_Ver(trainingData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
% Input:
%   trainingData: a table containing the same predictor and response
%   columns as imported into the app.
%
% Output:
%   trainedClassifier: a struct containing the trained classifier. The
%   struct contains various fields with information about the trained
%   classifier.
%
%   trainedClassifier.predictFcn: a function to make predictions on new
%   data.
%
%   validationAccuracy: a double containing the accuracy in percent. In
%   the app, the History list displays this overall accuracy score for
%   each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your original
% data or new data as the input argument trainingData.
%
% For example, to retrain a classifier trained with the original data set
% T, enter:
% [trainedClassifier, validationAccuracy] = trainClassifier(T)
%
% To make predictions with the returned 'trainedClassifier' on new data T2,
% use
% yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedClassifier.HowToPredict
%
% Auto-generated by MATLAB on 05-Feb-2019 14:02:34
%
% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'Cl', 'SO4', 'HCO3', 'pH', 'T'};
predictors = inputTable(:, predictorNames);
```

A5 trainRegressionModel_Eb_Boosted_Tree_Ver function

```
function [trainedModel, validationRMSE] =
trainRegressionModel_Eb_Boosted_Tree_Ver(trainingData)
% [trainedModel, validationRMSE] = trainRegressionModel(trainingData)
% returns a trained regression model and its RMSE. This code recreates the
% model trained in Regression Learner app. Use the generated code to
% automate training the same model with new data, or to learn how to
% programmatically train models.
%
% Input:
%   trainingData: a table containing the same predictor and response
%   columns as imported into the app.
%
% Output:
%   trainedModel: a struct containing the trained regression model. The
%   struct contains various fields with information about the trained
%   model.
%
%   trainedModel.predictFcn: a function to make predictions on new data.
%
%   validationRMSE: a double containing the RMSE. In the app, the
%   History list displays the RMSE for each model.
%
% Use the code to train the model with new data. To retrain your model,
% call the function from the command line with your original data or new
% data as the input argument trainingData.
%
% For example, to retrain a regression model trained with the original data
% set T, enter:
% [trainedModel, validationRMSE] = trainRegressionModel(T)
%
% To make predictions with the returned 'trainedModel' on new data T2, use
% yfit = trainedModel.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedModel.HowToPredict

% Auto-generated by MATLAB on 05-Feb-2019 14:07:43

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'Cl', 'SO4', 'HCO3', 'pH', 'T'};
predictors = inputTable(:, predictorNames);
response = inputTable.Eb;
isCategoricalPredictor = [false, false, false, false, false];

% Train a regression model
% This code specifies all the model options and trains the model.
template = templateTree(...
    'MinLeafSize', 8);
regressionEnsemble = fitensemble(...
    predictors, ...
    response, ...
    'Method', 'LSBoost', ...
```

```

'NumLearningCycles', 30, ...
'Learners', template, ...
'LearnRate', 0.1);

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
ensemblePredictFcn = @(x) predict(regressionEnsemble, x);
trainedModel.predictFcn = @(x) ensemblePredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedModel.RequiredVariables = {'Cl', 'SO4', 'HCO3', 'pH', 'T'};
trainedModel.RegressionEnsemble = regressionEnsemble;
trainedModel.About = 'This struct is a trained model exported from Regression Learner
R2018b.';
trainedModel.HowToPredict = sprintf('To make predictions on a new table, T, use: \n
yfit = c.predictFcn(T) \nreplacing ''c'' with the name of the variable that is this
struct, e.g. ''trainedModel''. \n \nThe table, T, must contain the variables returned
by: \n c.RequiredVariables \nVariable formats (e.g. matrix/vector, datatype) must
match the original training data. \nAdditional variables are ignored. \n \nFor more
information, see <a href=""matlab:helpview(fullfile(docroot, ''stats'',
''stats.map''), ''appregression_exportmodeltoworkspace'')">How to predict using an
exported model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'Cl', 'SO4', 'HCO3', 'pH', 'T'};
predictors = inputTable(:, predictorNames);
response = inputTable.Eb;
isCategoricalPredictor = [false, false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedModel.RegressionEnsemble, 'kFold', 5);

% Compute validation predictions
validationPredictions = kfoldPredict(partitionedModel);

% Compute validation RMSE
validationRMSE = sqrt(kfoldLoss(partitionedModel, 'LossFun', 'mse'));

```

A6 trainRegressionModel_Erp_Boosted_Tree_Ver function

```

function [trainedModel, validationRMSE] =
trainRegressionModel_Erp_Boosted_Tree_Ver(trainingData)
% [trainedModel, validationRMSE] = trainRegressionModel(trainingData)
% returns a trained regression model and its RMSE. This code recreates the
% model trained in Regression Learner app. Use the generated code to
% automate training the same model with new data, or to learn how to
% programmatically train models.
%
% Input:
%   trainingData: a table containing the same predictor and response
%   columns as imported into the app.

```

```

%
% Output:
%   trainedModel: a struct containing the trained regression model. The
%   struct contains various fields with information about the trained
%   model.
%
%   trainedModel.predictFcn: a function to make predictions on new data.
%
%   validationRMSE: a double containing the RMSE. In the app, the
%   History list displays the RMSE for each model.
%
% Use the code to train the model with new data. To retrain your model,
% call the function from the command line with your original data or new
% data as the input argument trainingData.
%
% For example, to retrain a regression model trained with the original data
% set T, enter:
% [trainedModel, validationRMSE] = trainRegressionModel(T)
%
% To make predictions with the returned 'trainedModel' on new data T2, use
% yfit = trainedModel.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedModel.HowToPredict

% Auto-generated by MATLAB on 02-Nov-2018 17:14:03

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'Cl', 'SO4', 'HCO3', 'pH', 'T'};
predictors = inputTable(:, predictorNames);
response = inputTable.Erp;
isCategoricalPredictor = [false, false, false, false, false];

% Train a regression model
% This code specifies all the model options and trains the model.
template = templateTree(...
    'MinLeafSize', 8);
regressionEnsemble = fitensemble(...
    predictors, ...
    response, ...
    'Method', 'LSBoost', ...
    'NumLearningCycles', 30, ...
    'Learners', template, ...
    'LearnRate', 0.1);

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
ensemblePredictFcn = @(x) predict(regressionEnsemble, x);
trainedModel.predictFcn = @(x) ensemblePredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedModel.RequiredVariables = {'Cl', 'SO4', 'HCO3', 'pH', 'T'};
trainedModel.RegressionEnsemble = regressionEnsemble;

```

```

trainedModel.About = 'This struct is a trained model exported from Regression Learner
R2018b.';
trainedModel.HowToPredict = sprintf('To make predictions on a new table, T, use: \n
yfit = c.predictFcn(T) \nreplacing ''c'' with the name of the variable that is this
struct, e.g. ''trainedModel''. \n \nThe table, T, must contain the variables returned
by: \n c.RequiredVariables \nVariable formats (e.g. matrix/vector, datatype) must
match the original training data. \nAdditional variables are ignored. \n \nFor more
information, see <a href="matlab:helpview(fullfile(docroot, ''stats'',
''stats.map''), ''appregression_exportmodeltoworkspace'')">How to predict using an
exported model</a>.'.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'Cl', 'SO4', 'HCO3', 'pH', 'T'};
predictors = inputTable(:, predictorNames);
response = inputTable.Erp;
isCategoricalPredictor = [false, false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedModel.ReggressionEnsemble, 'KFold', 5);

% Compute validation predictions
validationPredictions = kfoldPredict(partitionedModel);

% Compute validation RMSE
validationRMSE = sqrt(kfoldLoss(partitionedModel, 'LossFun', 'mse'));

```