Analysis of a Quantum Error Correcting Code using Quantum Process Calculus

Timothy A. S. Davidson
Department of Computer Science
University of Warwick, UK

Simon J. Gay
School of Computing Science
University of Glasgow, UK

Rajagopal Nagarajan*
Department of Computer Science
University of Warwick, UK

Ittoop Vergheese Puthoor**
School of Computing Science and
School of Physics and Astronomy
University of Glasgow, UK

We describe the use of quantum process calculus to describe and analyze quantum communication protocols, following the successful field of *formal methods* from classical computer science. The key idea is to define two systems, one modelling a protocol and one expressing a specification, and prove that they are *behaviourally equivalent*. We summarize the necessary theory in the process calculus CQP, including the crucial result that equivalence is a *congruence*, meaning that it is preserved by embedding in any context. We illustrate the approach by analyzing two versions of a quantum error correction system.

1 Introduction

Quantum process calculus is a generic term for a class of formal languages with which to describe and analyze the behaviour of systems that combine quantum and classical computation and communication. Quantum process calculi have been developed as part of a programme to transfer ideas from the field of *formal methods*, well established within classical computer science, to quantum systems. The field of formal methods provides theories, methodologies and tools for verifying the correctness of computing systems, usually systems that involve concurrent, communicating components. The motivation for developing quantum formal methods is partly to provide a conceptual understanding of concurrent, communicating quantum systems, and partly to support the future development of tools for verifying the correctness of practical quantum technologies such as cryptosystems.

Our own approach is based on a particular quantum process calculus called Communicating Quantum Processes (CQP), developed by Gay and Nagarajan [5]. Recent work on CQP has addressed the question of defining *behavioural equivalence* between processes, which formalizes the idea of observational indistinguishability. The aim is to support the following methodology for proving correctness of a system. First, define *System*, a process that models the system of interest. Second, define *Specification*, a simpler process that directly expresses the desired behaviour of *System*. Third, prove that these two processes are equivalent, meaning indistinguishable by any observer: $\mathit{System} \cong \mathit{Specification}$. This approach works best when the notion of equivalence is a *congruence*, meaning that it is preserved by inclusion in any environment. While there have been several attempts to define a congruence for a quantum process calculus, the problem has only recently been solved: for CQP, reported in Davidson's PhD thesis [2], and independently for qCCS by Feng *et al.* [4].

*Partially supported by "Process Algebra Approach to Distributed Quantum Computation and Secure Quantum Communication", Australian Research Council Discovery Project DP110103473.

**Supported by a Lord Kelvin / Adam Smith Scholarship from the University of Glasgow.

The present paper begins in Section 2 by reviewing the language of CQP and illustrating it with a model of a quantum error correcting code. Section 3 then summarizes the theory of behavioural equivalence for CQP, which has not previously been published other than in Davidson's thesis, and applies it to the error correcting code. Section 4 analyzes the system in the presence of errors that cannot be corrected. Finally, Section 5 concludes with an indication of directions for future work. The contributions of the paper are the first publication of the definition of a congruence for CQP, and the application of CQP congruence to examples beyond the teleportation and superdense coding systems that have been considered previously.

Related Work Lalire [10] defined a probabilistic branching bisimilarity for the process calculus QPAlg, based on the branching bisimilarity of van Glabbeek and Weijland [9], but it was not preserved by parallel composition. Feng et al. [3] developed qCCS and defined strong and weak probabilistic bisimilarity. Their equivalences are preserved by parallel composition with processes that do not change the quantum context. A later version of qCCS [17] excluded classical information and introduced the notion of approximate bisimilarity as a way of quantifying differences in purely quantum process behaviour. Their strong reduction-bisimilarity is a congruence is not sufficient for the analysis of most interesting quantum protocols, as the language does not include a full treatment of measurement. In recent work, Feng et al. [4] define a new version of qCCS and prove that weak bisimilarity is a congruence. They apply their result to quantum teleportation and superdense coding. The details of their equivalence relation are different from our full probabilistic branching bisimilarity, and a thorough comparison awaits further work.

The work presented in this paper contrasts with previous work on model-checking quantum systems. The QMC (Quantum Model-Checker) system [7, 8] is able to verify that a quantum protocol satisfies a specification expressed in a quantum logic, by exhaustively simulating every branch of its behaviour. The use of logical formulae is known as *property-oriented* specification, in distinction to the *process-oriented* specifications considered in the present paper. Because of the need for efficient simulation, QMC uses the stabilizer formalism [1] and is limited to Clifford group operations. Nevertheless, this is sufficient for the analysis of a simple error correcting code, and such an analysis appears in [8]. There are two main advantages of the process calculus approach. First, because we are using pen-and-paper reasoning rather than computational simulation, there is no restriction to stabilizer states. Second, the fact that equivalence is a congruence means that we can use equational reasoning to deduce further equivalences, whereas in the model-checking approach we only obtain the particular fact that is checked. The disadvantage of the process calculus approach is that, unlike the situation for classical process calculus, equivalence-checking has not yet been automated.

2 Communicating Quantum Processes (CQP)

CQP [5] is a process calculus for formally defining the structure and behaviour of systems that combine quantum and classical communication and computation. It is based on pi-calculus [12, 13], with the addition of primitive operations for quantum information processing. The general picture is that a system consists of a number of independent components, or *processes*, which can communicate by sending data along *channels*. In particular, qubits can be transmitted on channels. One of the distinctive features of CQP is its type system, which ensures that operations can only be applied to data of the appropriate type. The type system is also used to enforce the view of qubits as physical resources, each of which has a unique owning process at any given time. If a qubit is send from A to B, then ownership is transferred and *A* can no longer access it. Although typing is important, we will not discuss it in detail in the present paper; however, our CQP definitions will include type information because it usually forms useful documentation. Also, in the present paper, we will not give a full formal definition of the CQP language. Instead, in the next section, we will explain it informally in relation to our first model of a quantum error correction system.

2.1 Error Correction: A First Model

Our model of a quantum error correction system consists of three processes: *Alice*, *Bob* and *Noise*. *Alice* wants to send a qubit to *Bob* over a noisy channel, represented by *Noise*. She uses a simple error correcting code based on threefold repetition [14, Chapter 10]. This code is able to correct a single bit-flip error in each block of three transmitted qubits, so for the purpose of this example, in each block of three qubits, *Noise* either applies X to one of them or does nothing. *Bob* uses the appropriate decoding procedure to recover *Alice*'s original qubit. The CQP definition of *Alice* is as follows.

$$\begin{array}{l} \mathit{Alice}(a{:}\,\widehat{\ }[\mathsf{Qbit}], b{:}\,\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}]) = \\ \quad (\mathsf{qbit}\ y,z)a?[x{:}\,\mathsf{Qbit}]\,.\,\{x,z \mathrel{*}= \mathsf{CNot}\}\,.\,\{x,y \mathrel{*}= \mathsf{CNot}\}\,.\,b![x,y,z]\,.\,\mathbf{0} \end{array}$$

Alice is parameterized by two channels, a and b. In order to give *Alice* a general definition independent of the qubit to be sent to *Bob*, she will receive the qubit on channel a. The type of a is $\widehat{\ }[\mathsf{Qbit}]$, which is the type of a channel on which each message is a qubit. Channel b is where *Alice* sends the encoded qubits. Each message on b consists of three qubits, as indicated by the type $\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}]$.

The right hand side of the definition specifies *Alice*'s behaviour. The first term, $(\mathsf{qbit}\ y,z)$, allocates two fresh qubits, each in state $|0\rangle$, and gives them the local names y and z. Then follows a sequence of terms separated by dots. This indicates temporal sequencing, from left to right. $a?[x{:}\,\mathsf{Qbit}]$ specifies that a qubit is received from channel a and given the local name x. The term $\{x,z \mathrel{*}= \mathsf{CNot}\}$ specifies that the CNot operation is applied to qubits x and z; the next term is similar. These operations implement the threefold repetition code: if the intial state of x is $|0\rangle$ (respectively, $|1\rangle$) then the state of x,y,z becomes $|000\rangle$ (respectively, $|111\rangle$). In general, of course, the initial state of x may be a superposition, and then so will be the final state of x,y,z. Finally, the term $b![x,y,z]$ means that the qubits x,y,z are sent as a message on channel b. The term $\mathbf{0}$ simply indicates termination.

We model a noisy quantum channel by the process *Noise*, which receives three qubits from channel b (connected to *Alice*) and sends three (possibly corrupted) qubits on channel c (connected to *Bob*). *Noise* has four possible actions: do nothing, or apply X to one of the three qubits. These actions are chosen with equal probability. We produce probabilistic behaviour by introducing fresh qubits in state $|0\rangle$, applying H to put them into state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and then measuring in the standard basis. The definition of *Noise* is split into two sub-processes, of which the first, *NoiseRnd*, produces two random classical bits and sends them to the second, *NoiseErr*, on channel p. This programming style, using internal messages instead of assignment to variables, is typical of pi-calculus.

$$\mathit{NoiseRnd}(p{:}\,\widehat{\ }[\mathsf{bit},\mathsf{bit}]) \quad = \quad (\mathsf{qbit}\ u,v)\{u \mathrel{*}= \mathsf{H}\}\,.\,\{v \mathrel{*}= \mathsf{H}\}\,.\,p![\mathsf{measure}\ u, \mathsf{measure}\ v]\,.\,\mathbf{0}$$

The process *NoiseErr* receives three qubits from channel b, and two classical bits from channel p. It interprets the classical bits, locally named j and k, as instructions for corrupting the qubits. This uses appropriate Boolean combinations of j and k to construct conditional quantum operations such as $\mathsf{X}^{j\bar{k}}$.

$$\begin{array}{l} \mathit{NoiseErr}(b{:}\,\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}], p{:}\,\widehat{\ }[\mathsf{bit},\mathsf{bit}], c{:}\,\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}]) = \\ \quad b?[x{:}\,\mathsf{Qbit}, y{:}\,\mathsf{Qbit}, z{:}\,\mathsf{Qbit}]\,.\,p?[j{:}\,\mathsf{bit}, k{:}\,\mathsf{bit}]\,.\,\{x \mathrel{*}= \mathsf{X}^{jk}\}\,.\,\{y \mathrel{*}= \mathsf{X}^{j\bar{k}}\}\,.\,\{z \mathrel{*}= \mathsf{X}^{\bar{j}k}\}\,.\,c![x,y,z]\,.\,\mathbf{0} \end{array}$$

The complete *Noise* process consists of *NoiseRnd* and *NoiseErr* in parallel, indicated by the vertical bar. Channel p is designated as a private local channel; this is specified by $(\mathsf{new}\ p)$. This construct comes from pi-calculus, where it can be used to dynamically create fresh channels, but here we are using it in the style of older process calculi such as CCS, to indicate a channel with restricted scope. Putting *NoiseRnd* and *NoiseErr* in parallel means that the output on p in *NoiseRnd* synchronizes with the input on p in *NoiseErr*, so that data is transferred.

$$\mathit{Noise}(b{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}],c{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}]) \quad = \quad (\mathsf{new}\ p)(\mathit{NoiseRnd}(p)\,|\,\mathit{NoiseErr}(b,p,c))$$

Bob consists of *BobRec* and *BobCorr*, where *BobRec* receives the qubits and measures the error syndrome, and *BobCorr* applies the appropriate correction. An internal channel p is used to transmit the result of the measurement, as well as the original qubits, again in pi-calculus style. After correcting the error in the group of three qubits, *BobCorr* reconstructs a quantum state in which qubit x has the original state received by *Alice* and is separable from the auxiliary qubits. Finally, *BobCorr* outputs x on channel d.

$$\begin{aligned}\mathit{BobRec}(c{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}],p{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit},\mathsf{bit},\mathsf{bit}]) &= (\mathsf{qbit}\ s,t)c?[x{:}\mathsf{Qbit},y{:}\mathsf{Qbit},z{:}\mathsf{Qbit}]\,.\\ \{x,s \mathrel{*}= \mathsf{CNot}\}\,.\,\{y,s \mathrel{*}= \mathsf{CNot}\}\,.\,\{x,t \mathrel{*}= \mathsf{CNot}\}\,.\,&\{z,t \mathrel{*}= \mathsf{CNot}\}\,.\,p![x,y,z,\mathsf{measure}\ s,\mathsf{measure}\ t]\,.\,\mathbf{0}\end{aligned}$$

$$\begin{aligned}\mathit{BobCorr}(p{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit},\mathsf{bit},\mathsf{bit}],d{:}\widehat{\ }[\mathsf{Qbit}]) &= p?[x{:}\mathsf{Qbit},y{:}\mathsf{Qbit},z{:}\mathsf{Qbit},j{:}\mathsf{bit},k{:}\mathsf{bit}]\,.\\ \{x \mathrel{*}= \mathsf{X}^{jk}\}\,.\,\{y \mathrel{*}= \mathsf{X}^{j\bar{k}}\}\,.\,\{z \mathrel{*}= \mathsf{X}^{\bar{j}k}\}\,.\,&\{x,y \mathrel{*}= \mathsf{CNot}\}\,.\,\{x,z \mathrel{*}= \mathsf{CNot}\}\,.\,d![x]\,.\,\mathbf{0}\end{aligned}$$

$$\mathit{Bob}(c{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}],d{:}\widehat{\ }[\mathsf{Qbit}]) = (\mathsf{new}\ p)(\mathit{BobRec}(c,p)\,|\,\mathit{BobCorr}(p,d))$$

The overall effect of the error correcting system is to input a qubit from channel a and output a qubit, in the same state, on channel d, in the presence of noise. The complete system is defined as follows.

$$\mathit{QECC}(a{:}\widehat{\ }[\mathsf{Qbit}],d{:}\widehat{\ }[\mathsf{Qbit}]) = (\mathsf{new}\ b,c)(\mathit{Alice}(a,b)\,|\,\mathit{Noise}(b,c)\,|\,\mathit{Bob}(c,d))$$

When we consider correctness of the error correction system, we will prove that *QECC* is equivalent to the following *identity process*, which by definition transmits a single qubit faithfully.

$$\mathit{Identity}(a{:}\widehat{\ }[\mathsf{Qbit}],d{:}\widehat{\ }[\mathsf{Qbit}]) = a?[x{:}\mathsf{Qbit}]\,.\,d![x]\,.\,\mathbf{0}$$

2.2 Semantics of CQP

The intended behaviour of the processes in the error correction system was described informally in the previous section, but in fact the behaviour is precisely specified by the formal semantics of CQP. In this section we will explain the formal semantics, although without giving all of the definitions. Full details can be found in Davidson's PhD thesis [2].

In classical process calculus, the semantics is defined by labelled transitions between syntactic process terms. For example, a process of the form $c![2]\,.\,P$, where P is some continuation process, has the transition

$$c![2]\,.\,P \xrightarrow{c![2]} P. \qquad (1)$$

The label $c![2]$ indicates the potential interaction of the process with the environment. In order for this potential interaction to become an actual step in the behaviour of a system, there would have to be another process, ready to receive on channel c. A suitable process is $c?[x]\,.\,Q$, where Q is some continuation process. The labelled transition representing the potential input is

$$c?[x]\,.\,Q \xrightarrow{c?[v]} Q\{v/x\}. \qquad (2)$$

Here v stands for any possible input value, and $Q\{v/x\}$ means Q with the value v substituted for the variable x. If these two processes are put in parallel then each has a partner for its potential interaction, and the input and output can synchronize, resulting in a τ transition which represents a single step of behaviour:

$$c![2]\,.\,P \mid c?[x]\,.\,Q \xrightarrow{\tau} P \mid Q\{2/x\}.$$

The complete definition of the semantics takes the form of a collection of labelled transition rules. Transition (1) becomes a general rule for output if the value 2 is replaced by a general value v. Transition (2) is a general rule for input. The interaction between input and output is defined by the rule

$$\frac{P \xrightarrow{c![v]} P' \qquad Q \xrightarrow{c?[v]} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

which specifies that if the transitions above the line (hypotheses) are possible then so is the transition below the line (conclusion). Full details of this style of semantics, in relation to pi-calculus, can be found in [12, 15].

To define the semantics of a quantum process calculus such as CQP, we need to include a representation of the quantum state. Because of entanglement, the quantum state is a global property. It also turns out to be necessary to specify which qubits in the global quantum state are owned by (i.e. accessible to) the process term under consideration. We work with *configurations* such as

$$([q,r \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; q; c![q]\,.\,P). \tag{3}$$

This configuration means that the global quantum state consists of two qubits, q and r, in the specified state; that the process term under consideration has access to qubit q but not to qubit r ; and that the process itself is $c![q]\,.\,P$. Now consider a configuration with the same quantum state but a different process term:

$$([q,r \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; r; d![r]\,.\,Q).$$

The parallel composition of these configurations is the following:

$$([q,r \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; q,r; c![q]\,.\,P \mid d![r]\,.\,Q)$$

where the quantum state is still the same.

The semantics of CQP consists of labelled transitions between configurations, which are defined in a similar way to classical process calculus. For example, configuration (3) has the transition

$$([q,r \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; q; c![q]\,.\,P) \xrightarrow{c![q]} ([q,r \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; \emptyset; P).$$

The quantum state is not changed by this transition, but because qubit q is output, the continuation process P no longer has access to it; the final configuration has an empty list of owned qubits.

Previous papers on CQP [5, 6] defined the semantics in a different style. Instead of labelled transitions there were *reductions*, corresponding to τ transitions, and these were defined directly. However, although reduction semantics allows the behaviour of a complete system to be defined, labelled transitions and their interpretation as *potential* interactions are necessary in order to define equivalence between processes, which is the focus of the present paper.

As well as the different style of definition used in previous work, there is a very significant difference in the way that the semantics treats quantum measurement. In the original reduction semantics of CQP, a measurement leads to a probability distribution over configurations, which at the next step reduces probabilistically to one particular configuration. But in order for equivalence of processes to have the crucial property of *congruence*, the semantics must incorporate a more sophisticated analysis of measurement, in which *mixed configurations* play an essential role.

If the result of a quantum measurement is not made available to an observer then the system is considered to be in a mixed state, but it is not sufficient to simply write a mixed quantum state in a configuration. In general the mixture includes the process term, because the measurement result occurs within the term.

Example 1 $([q \mapsto \alpha_0|0\rangle + \alpha_1|1\rangle]; q; c![\mathsf{measure}\ q].P) \xrightarrow{\tau} \oplus_{i\in\{0,1\}} |\alpha_i|^2\ ([q \mapsto |i\rangle]; q; \lambda x \bullet c![x].P; i)$.

This transition represents the effect of a measurement, within a process which is going to output the result of the measurement; the output, however, is not part of the transition, which is why it is a τ transition and the process term on the right still contains $c![]$. The configuration on the left is a *pure configuration*, as described before. On the right we have a *mixed configuration* in which the $\oplus$ ranges over the possible outcomes of the measurement and the $|\alpha_i|^2$ are the weights of the components in the mixture. The quantum state $[q \mapsto |i\rangle]$ corresponds to the measurement outcome. The expression $\lambda x \bullet c![x].P$ is not a λ-calculus function, but represents the fact that the components of the mixed configuration have the same process structure and differ only in the values corresponding to measurement outcomes. The final term in the configuration, i, shows how the abstracted variable x should be instantiated in each component. Thus the λx represents a term into which expressions may be substituted, which is the reason for the λ notation. So the mixed configuration is essentially an abbreviation of

$$|\alpha_0|^2([q \mapsto |0\rangle]; q; c![0].P\{0/x\}) \oplus |\alpha_1|^2([q \mapsto |1\rangle]; q; c![1].P\{1/x\}).$$

If a measurement outcome is output then it becomes apparent to an observer which of the possible states the system is in. This is represented by probabilistic branching, after which we consider that system to be in one branch or the other — it is no longer a mixture of the two. Example 2 shows the effect of the output from the final configuration of Example 1. The output transition produces the intermediate configuration, which is a probability distribution over pure configurations (in contrast to a mixed configuration; note the change from $\oplus$ to $\boxplus$). Because it comes from a mixed configuration, the output transition contains a *set* of possible values. From the intermediate configuration there are two possible probabilistic transitions, of which one is shown ($\overset{|\alpha_0|^2}{\rightsquigarrow}$).

Example 2

$$\oplus_{i\in\{0,1\}} |\alpha|^2_i\ ([q \mapsto |i\rangle]; q; \lambda x \bullet c![x].P; i) \xrightarrow{c![\{0,1\}]}$$
$$\boxplus_{i\in\{0,1\}} |\alpha_i|^2([q \mapsto |i\rangle]; q; \lambda x \bullet P; i) \overset{|\alpha_0|^2}{\rightsquigarrow} ([q \mapsto |0\rangle]; q; \lambda x \bullet P; 0)$$

Measurement outcomes may be communicated between processes without creating a probability distribution. In these cases an observer must still consider the system to be in a mixed configuration. In Example 3 there is a mixed configuration on the left, with arbitrary weights g_i, which we imagine to have been produced by a measurement. However, there is now a receiver for the output. Although there is no difference in process Q between the two components of the mixed configuration, we include it in the λ because the communication will propagate the different possible values for x to Q.

Example 3

$$\oplus_{i\in\{0,1\}}\, g_i\,([q\mapsto|i\rangle];q;\lambda x\bullet(c![x].P\parallel c?[y].Q);i)\xrightarrow{\tau}\oplus_{i\in\{0,1\}}\, g_i\,([q\mapsto|i\rangle];q;\lambda x\bullet(P\parallel Q\{x/y\});i)$$

The full definition of the labelled transition semantics covers more complex possibilities. For example, if incomplete information about a measurement is revealed, the resulting configuration is in general a probability distribution over mixed configurations. The aspects of the semantics that are relevant to the present paper will be illustrated further in relation to the error correction example. Now we define some notation.

There are two types of transition: probabilistic transitions which take the form $\boxplus_i p_i s_i \overset{p_i}{\rightsquigarrow} s_i$ where $\forall i.(p_i<1)$, and non-deterministic transitions which have the general form $s\xrightarrow{\alpha}\boxplus_i p_i s_i$ where $\forall i.(p_i\leq 1)$ and α is an *action*. The notation $\boxplus_i p_i s_i \equiv p_1\bullet s_1\boxplus\cdots\boxplus\ p_n\bullet s_n$ denotes a probability distribution over configurations in which $\sum_i p_i=1$. If there is only a single configuration (with probability 1) we omit the probability, for example $s\xrightarrow{\alpha}s'$.

The separation of probabilistic and non-deterministic transitions avoids the need to consider non-deterministic and probabilistic transitions from the same configuration. The relations $\xrightarrow{\alpha}$ and $\overset{\pi}{\rightsquigarrow}$ induce a partition of the set $\mathscr{S}$ of configurations into non-deterministic configurations $\mathscr{S}_n$ and probabilistic configurations $\mathscr{S}_p$: let $\mathscr{S}_p=\{s\in\mathscr{S}\mid\exists\pi\in(0,1],\exists t\in\mathscr{S},s\overset{\pi}{\rightsquigarrow}t\}$; and let $\mathscr{S}_n=\mathscr{S}\setminus\mathscr{S}_p$. By this definition a configuration with no transitions belongs to $\mathscr{S}_n$. This notation will be used in Section 3.

2.3 Execution of QECC

We show the interesting steps in one possible execution of QECC, omitting the new declarations from the process terms to reduce clutter. The semantics of CQP is non-deterministic, so transitions can proceed in a different order; the order shown here is chosen for presentational convenience. The initial configuration is $(\emptyset;\emptyset;Alice\,|\,Noise\,|\,Bob)$. In the first few steps, the processes execute qbit terms, constructing a global quantum state:

$$([y,z,u,v,s,t\mapsto|000000\rangle];y,z,u,v,s,t;Alice'\,|\,Noise'\,|\,Bob')$$

Alice receives qubit x, in state $\alpha|0\rangle+\beta|1\rangle$, from the environment, via transition $\xrightarrow{a?[x]}$, which expands the quantum state. We now abbreviate the list of qubits to $\widetilde{q}=x,y,z,u,v,s,t$. After some τ transitions corresponding to *Alice*'s CNot operations, we have:

$$([\widetilde{q}\mapsto\alpha|0000000\rangle+\beta|1110000\rangle];\widetilde{q};b![x,y,z]\,.\,\mathbf{0}\,|\,Noise'\,|\,Bob')$$

$Noise'=NoiseErr\,|\,NoiseRnd'$ ($NoiseRnd'$ has already done its qbit). The output on b interacts with the input on b in *NoiseErr*. Meanwhile, the measurements in *NoiseRnd* produce a mixed configuration because the results are communicated internally, to *NoiseErr*:

$$\oplus_{j,k\in\{0,1\}}\tfrac{1}{4}([\widetilde{q}\mapsto\alpha|000jk00\rangle+\beta|111jk00\rangle];\widetilde{q};$$
$$\lambda jk\bullet\{x\mathrel{*}=\mathsf{X}^{jk}\}\,.\,\{y\mathrel{*}=\mathsf{X}^{j\bar{k}}\}\,.\,\{z\mathrel{*}=\mathsf{X}^{\bar{j}k}\}\,.\,c![x,y,z]\,.\,\mathbf{0}\,|\,Bob';j,k)$$

After τ transitions from the controlled X operations, we can write the mixed configuration explicitly:

$$\begin{array}{l}\tfrac{1}{4}([\widetilde{q}\mapsto\alpha|0000000\rangle+\beta|1110000\rangle];\widetilde{q};c![x,y,z]\,.\,\mathbf{0}\,|\,Bob')\\ \oplus\tfrac{1}{4}([\widetilde{q}\mapsto\alpha|0010100\rangle+\beta|1100100\rangle];\widetilde{q};c![x,y,z]\,.\,\mathbf{0}\,|\,Bob')\\ \oplus\tfrac{1}{4}([\widetilde{q}\mapsto\alpha|0101000\rangle+\beta|1011000\rangle];\widetilde{q};c![x,y,z]\,.\,\mathbf{0}\,|\,Bob')\\ \oplus\tfrac{1}{4}([\widetilde{q}\mapsto\alpha|1001100\rangle+\beta|0111100\rangle];\widetilde{q};c![x,y,z]\,.\,\mathbf{0}\,|\,Bob')\end{array}$$

The remaining transitions operate within the mixed configuration. In each component of the mixture, the measurement of s,t by *BobRec* has a deterministic outcome, so no further mixedness is introduced. Eventually we have a mixed configuration in which the process term is the same, $d![x].\mathbf{0}$, in every component, so we can just consider the mixed *state*, which is

$$\oplus_{j,k\in\{0,1\}}\frac{1}{4}[x,y,z,u,v,s,t \mapsto \alpha|000jkjk\rangle+\beta|100jkjk\rangle].$$

The mixture over j,k is the residue of the random choice made by *NoiseRnd*, and the dependence of s and t on j,k is because *BobRec*'s measurement recovers the values of j and k (which is what allows the error to be corrected). In this final mixed state, the reduced density matrix of x, which is what we are interested in when x is output, is the same as the original density matrix of x.

3 Behavioural Equivalence of CQP Processes

The process calculus approach to verification is to define a process *System* which models the system of interest, another process *Spec* which expresses the specification that *System* should satisfy, and then prove that *System* and *Spec* are equivalent. Usually *Spec* is defined in a sufficiently simple way that it can be taken as self-evident that it accurately represents the desired specification.

What do we mean by *equivalent*? The idea is that two processes are equivalent if their behaviour is indistinguishable by an observer. That is, if they do the same thing in the same circumstances. Equivalence relations in this style are generically called *behavioural* equivalences. Suppose that $\cong$ is an equivalence relation on processes. The ideal situation is for $\cong$ to have a further property called *congruence*, which means that it is preserved by all of the constructs of the process calculus. A convenient way to express this property involves the notion of a *process context* $C[]$. This is a process term containing a *hole*, represented by $[]$, into which a process term may be placed. For example, $c?[x].[]$ is a context, and putting the process $d![x].\mathbf{0}$ into the hole results in the process $c?[x].d![x].\mathbf{0}$.

Definition 1 *An equivalence relation $\cong$ on processes is a* congruence *if*

$$\forall P,Q.\ P\cong Q \Rightarrow \forall C[].\ C[P]\cong C[Q].$$

This definition of congruence corresponds to the idea that observers are themselves expressed as processes. Congruence, in addition to the property of being an equivalence relation, is what is required in order to allow equational reasoning about equivalence of processes. It means that if a system satisfies its specification, then it continues to satisfy its specification no matter what environment it is placed in.

From the beginning of the study of quantum process calculus, the aim was to define a behavioural equivalence with the congruence property. This was not straightforward and took several years to achieve; Lalire [10] describes an unsuccessful attempt. Recently the congruence problem has been solved by the first three authors of the present paper [2] for CQP and, independently, by Feng *et al.* [4] for qCCS.

We will now present the concept of *bisimilarity*, which is the main approach to behavioural equivalence, and then define a particular form of bisimilarity, called *probabilistic branching bisimilarity*, which is a congruence for CQP.

3.1 Strong Bisimilarity

The basic idea of bisimilarity is that if two processes are equivalent then any labelled transition by one can be matched by the other, and the resulting processes are again equivalent. It is worth presenting the definition of the prototypical example, *strong bisimilarity* [11], as a model for later definitions. The most general setting for the definition is to consider a *labelled transition system*, which consists of a set of states and a three-place relation on $States \times Labels \times States$, written $s \xrightarrow{\alpha} t$. A labelled transition system can be regarded as a labelled directed graph whose vertices are the states. We will consider relations on the set of states. The definition of strong bisimilarity proceeds in two stages. First we define the property of *strong bisimulation*, which a particular relation might or might not have.

Definition 2 (Strong Bisimulation) *A relation* $\mathscr{R}$ *is a* strong bisimulation *if whenever* $(P,Q) \in \mathscr{R}$ *then for all labels* α, *both*

1. *if* $P \xrightarrow{\alpha} P'$ *then* $Q \xrightarrow{\alpha} Q'$ *and* $(P',Q') \in \mathscr{R}$, *and*
2. *if* $Q \xrightarrow{\alpha} Q'$ *then* $P \xrightarrow{\alpha} P'$ *and* $(P',Q') \in \mathscr{R}$.

For a given labelled transition system there are many relations that have the property of strong bisimulation, including (trivially) the empty relation. The key idea is to define *strong bisimilarity* to be the union of all strong bisimulations, or equivalently the largest strong bisimulation. In other words, P and Q are strong bisimilar (denoted $P \sim Q$) if and only if there exists a bisimulation $\mathscr{R}$ such that $(P,Q) \in \mathscr{R}$.

3.2 Probabilistic Branching Bisimilarity

One of the characteristics of strong bisimilarity is that it is a stronger relation than trace equivalence; it is possible for two processes to generate the same sequences of labels, but not be strong bisimilar. Strong bisimilarity depends on the branching structure of the processes as well as on their sequences of labels. Another characteristic is that *every* transition must be matched exactly, including τ transitions. However, because they arise from internal communications, it is often undesirable to insist that equivalent processes must match each other's τ transitions. Hence weaker variations of bisimilarity have been defined, including *weak bisimilarity* [11], which ignores τ transitions, and *branching bisimilarity* [9], which reduces the significance of τ transitions but retains information about their branching structure.

When considering equivalences for quantum process calculus, it is necessary to take probability into account; even with the treatment of mixed configurations described in Section 2, there is probabilistic behaviour when measurement results are revealed to the observer. There are several varieties of probabilistic bisimilarity for classical probabilistic process calculi, including *probabilistic branching bisimilarity* [16]. The equivalence for CQP defined by Davidson [2], which turns out to be a congruence, is a form of probabilistic branching bisimilarity, adapted to the situation in which probabilistic behaviour comes from quantum measurement. A key point is that when considering matching of input or output transitions involving qubits, it is the reduced density matrices of the transmitted qubits that are required to be equal.

Although we did not present the full definition of the labelled transition semantics for CQP, we will now define probabilistic branching bisimilarity in full. In Section 3.4, the definition will be applied to the error correction example. The definitions in the remainder of this section are from Davidson's thesis [2].

Notation: Let $\xrightarrow{\tau}^{+}$ denote zero or one τ transitions; let $\Longrightarrow$ denote zero or more τ transitions; and let $\xRightarrow{\alpha}$ be equivalent to $\Longrightarrow\xrightarrow{\alpha}\Longrightarrow$. We write $\widetilde{q}$ for a list of qubit names, and similarly for other lists.

Definition 3 (Density Matrix of Configurations) *Let* $\sigma_i = [\widetilde{p} \mapsto |\psi_i\rangle]$ *and* $\widetilde{q} \subseteq \widetilde{p}$ *and* $s_i = (\sigma_i;\omega;\lambda\widetilde{x} \bullet P;\widetilde{v_i})$ *and* $s = \oplus_i\, g_i\, s_i$. *Then*

1. $\rho(\sigma_i) = |\psi_i\rangle\langle\psi_i|$
2. $\rho^{\widetilde{q}}(\sigma_i) = \mathrm{tr}_{\widetilde{p}\setminus\widetilde{q}}(|\psi_i\rangle\langle\psi_i|)$
3. $\rho(s_i) = \rho(\sigma_i)$
4. $\rho^{\widetilde{q}}(s_i) = \rho^{\widetilde{q}}(\sigma_i)$
5. $\rho(s) = \sum_i g_i \rho(s_i)$
6. $\rho^{\widetilde{q}}(s) = \sum_i g_i \rho^{\widetilde{q}}(s_i)$

We also introduce the notation ρ_E to denote the reduced density matrix of the *environment* qubits. Formally, if $s = ([\widetilde{q} \mapsto |\psi\rangle]; \widetilde{p}; P)$ then $\rho_E(s) = \rho^{\widetilde{r}}(s)$ where $\widetilde{r} = \widetilde{q} \setminus \widetilde{p}$. The definition of ρ_E is extended to mixed configurations in the same manner as ρ.

Again let $\mathscr{S}$ be the set of configurations. The probabilistic function $\mu : \mathscr{S} \times \mathscr{S} \to [0,1]$ is defined in the style of [16]. It allows non-deterministic transitions to be treated as transitions with probability 1, which is necessary when calculating the total probability of reaching a terminal state. $\mu(s,t) = \pi$ if $s \overset{\pi}{\rightsquigarrow} t$; $\mu(s,t) = 1$ if $s = t$ and $s \in \mathscr{S}_n$; $\mu(s,t) = 0$ otherwise.

Definition 4 (Probabilistic Branching Bisimulation) *An equivalence relation $\mathscr{R}$ on configurations is a* probabilistic branching bisimulation *on configurations if whenever $(s,t) \in \mathscr{R}$ the following conditions are satisfied.*

I. If $s \in \mathscr{S}_n$ and $s \xrightarrow{\tau} s'$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{\tau}{}^{+} t''$ with $(s,t') \in \mathscr{R}$ and $(s',t'') \in \mathscr{R}$.

II. If $s \xrightarrow{c![V,\widetilde{q}_1]} s'$ where $s' = \boxplus_{j \in \{1\ldots m\}} p_j s'_j$ and $V = \{\widetilde{v}_1, \ldots, \widetilde{v}_m\}$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{c![V,\widetilde{q}_2]} t''$ with

a) $(s,t') \in \mathscr{R}$,

b) $t'' = \boxplus_{j \in \{1\ldots m\}} p_j t''_j$,

c) for each $j \in \{1,\ldots,m\}$, $\rho_E(s'_j) = \rho_E(t''_j)$.

d) for each $j \in \{1,\ldots,m\}$, $(s'_j, t''_j) \in \mathscr{R}$.

III. If $s \xrightarrow{c?[\widetilde{v}]} s'$ then $\exists t', t''$ such that $t \Longrightarrow t' \xrightarrow{c?[\widetilde{v}]} t''$ with $(s,t') \in \mathscr{R}$ and $(s',t'') \in \mathscr{R}$.

IV. If $s \in \mathscr{S}_p$ then $\mu(s,D) = \mu(t,D)$ for all classes $D \in \mathscr{S}/\mathscr{R}$.

This relation follows the standard definition of branching bisimulation [9] with additional conditions for probabilistic configurations and matching quantum information. In condition II we require that the distinct set of values V must match and although the qubit names ($\widetilde{q}_1$ and $\widetilde{q}_2$) need not be identical, their respective reduced density matrices ($\rho^{\widetilde{q}_1}(s)$ and $\rho^{\widetilde{q}_2}(t')$) must.

Condition IV provides the matching on probabilistic configurations following the approach of [16]. In this relation, a probabilistic configuration which necessarily evolves from an output will satisfy IV if the prior configuration satisfies II d). It is necessary to include the latter condition to ensure that the probabilities are paired with their respective configurations.

Naturally this leads to the following definition of bisimilarity on configurations.

Definition 5 (Probabilistic Branching Bisimilarity) *Configurations s and t are* probabilistic branching bisimilar*, denoted $s \leftrightarroweq t$, if there exists a probabilistic branching bisimulation $\mathscr{R}$ such that $(s,t) \in \mathscr{R}$.*

What we really want is equivalence of processes, independently of configurations (i.e. independently of particular quantum states).

Definition 6 (Probabilistic Branching Bisimilarity of Processes) *Processes P and Q are* probabilistic branching bisimilar*, denoted $P \leftrightarroweq Q$, if and only if for all σ, $(\sigma; \emptyset; P) \leftrightarroweq (\sigma; \emptyset; Q)$.*

For convenience, in the remainder of this paper *bisimilarity* will refer to probabilistic branching bisimilarity and it will be clear from the context whether this is the relation on processes or configurations. The same symbol, $\leftrightarroweq$, is used for both relations.

It turns out that probabilistic branching bisimilarity is not a congruence because it is not preserved by substitution of values for variables, which is significant because of the use of substitution to define the semantics of input. We therefore define a stronger relation, *full probabilistic branching bisimilarity*, which is the closure of probabilistic branching bisimilarity under substitutions.

Definition 7 (Full probabilistic branching bisimilarity) *Processes P and Q are* full probabilistic branching bisimilar, *denoted $P \leftrightarroweq^c Q$, if for all substitutions κ and all quantum states σ, $(\sigma;\widetilde{q};P\kappa) \leftrightarroweq (\sigma;\widetilde{q};Q\kappa)$.*

We are now able to state the main result of [2].

Theorem 1 (Full probabilistic branching bisimilarity is a congruence) *If $P \leftrightarroweq^c Q$ then for any context $C[]$, if $C[P]$ and $C[Q]$ are typable then $C[P] \leftrightarroweq^c C[Q]$.*

The condition that $C[P]$ and $C[Q]$ are typable is used to ensure that the context does not manipulate qubits that are owned by P or Q.

3.3 Mixed Configurations and Congruence

A simple example will illustrate why the congruence result depends crucially on the use of mixed configurations. Consider the processes

$$\begin{array}{rcl} P(a{:}\widehat{\ }[\mathsf{Qbit}]) & = & a?[x{:}\mathsf{Qbit}]\,.\,\{\mathsf{measure}\ x\}\,.\,\mathbf{0} \\ Q(a{:}\widehat{\ }[\mathsf{Qbit}]) & = & a?[x{:}\mathsf{Qbit}]\,.\,\{x \mathrel{*}= \mathsf{H}\}\{\mathsf{measure}\ x\}\,.\,\mathbf{0} \end{array}$$

P and Q are probabilistic branching bisimilar, because in any quantum state they can match each other's transitions. For input transitions this is because they can both input a single qubit, and for output transitions it is trivial because neither process produces any output. The actions within each process produce τ transitions, which are absorbed into the input transitions according to the definition of probabilistic branching bisimulation.

Now consider P and Q in parallel with $R(b{:}\widehat{\ }[\mathsf{Qbit}]) = b![q]\,.\,\mathbf{0}$ in the quantum state $[p,q \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]$. That is, consider the configurations

$$([p,q \mapsto \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; p,q; P \mid R) \qquad ([p,q \mapsto \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle)]; p,q; Q \mid R)$$

The interesting situation is when the measurement in P or Q occurs before the output in R. Imagine, first, that the semantics of CQP handles the measurement by producing a probability distribution; recall that this is not the actual semantics of measurement. In $P \mid R$ the quantum state before the measurement is $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and the state after the measurement is either $|00\rangle$ or $|11\rangle$ with equal probability. The qubit output by R has reduced density matrix $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ or $\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$. In $Q \mid R$ the quantum state before the measurement is $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle)$ and the state after the measurement is either $\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$ or $\frac{1}{\sqrt{2}}(|10\rangle - |11\rangle)$ with equal probability. The qubit output by R has reduced density matrix $\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ or $\frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$. It is therefore impossible for $P \mid R$ and $Q \mid R$ to match each other's outputs.

Actually, of course, the semantics of CQP does not produce a probability distribution in this case, because the result of the measurement is not output. Instead, from $P \mid R$ we get the mixed configuration

$$\frac{1}{2}([p,q \mapsto |00\rangle]; p,q; \mathbf{0} \mid b![q]\,.\,\mathbf{0}) \oplus \frac{1}{2}([p,q \mapsto |11\rangle]; p,q; \mathbf{0} \mid b![q]\,.\,\mathbf{0}) \tag{4}$$

and from $Q \mid R$ we get the mixed configuration

$$\frac{1}{2}([p,q \mapsto \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)]; p,q; \mathbf{0} \mid b![q]\,.\,\mathbf{0}) \oplus \frac{1}{2}([p,q \mapsto \frac{1}{\sqrt{2}}(|10\rangle - |11\rangle)]; p,q; \mathbf{0} \mid b![q]\,.\,\mathbf{0}). \tag{5}$$

The calculation of the reduced density matrix of the qubit output by R, taking into account the contributions of each component of the mixed configuration, gives $\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$ in both cases. This enables $P\,|\,R$ and $Q\,|\,R$ to match each other's outputs, and in fact (although we do not show it here), it is straightforward to construct a probabilistic branching bisimulation relation containing $(P\,|\,R, Q\,|\,R)$.

3.4 Correctness of QECC

We now sketch the proof that $\mathit{QECC} \leftrightarrows^c \mathit{Identity}$, which by Theorem 1 implies that the error correction system works in any context. An interesting consequence is that the qubit being transmitted may be part of any quantum state, meaning that it is correctly transmitted with error correction even if it is entangled with other qubits; the entanglement is also preserved by the error correction system. This property of error correction, although easily verified by hand, is not usually stated explicitly in the literature.

Proposition 1 $\mathit{QECC} \leftrightarrows^c \mathit{Identity}$.

Proof: First we prove that $\mathit{QECC} \leftrightarrows \mathit{Identity}$, by defining an equivalence relation $\mathscr{R}$ that contains the pair $((\sigma;\emptyset;\mathit{QECC}),(\sigma;\emptyset;\mathit{Identity}))$ for all σ and is closed under their transitions. $\mathscr{R}$ is defined by taking its equivalence classes to be the $S_i(\sigma)$ defined below, for all states σ. The idea is to group configurations according to the sequences of observable transitions leading to them. S_2 is also parameterized by the input qubit, as this affects the output qubit and hence the equivalence class.

$$\begin{array}{rcl} S_1(\sigma) & = & \{s \mid (\sigma;\emptyset;P) \Longrightarrow s \text{ and } P \in \{\mathit{QECC},\mathit{Identity}\}\} \\ S_2(\sigma,x) & = & \{s \mid (\sigma;\emptyset;P) \overset{a?[x]}{\Longrightarrow} s \text{ and } P \in \{\mathit{QECC},\mathit{Identity}\}\} \\ S_3(\sigma) & = & \{s \mid (\sigma;\emptyset;P) \overset{a?[x]}{\Longrightarrow}\overset{d![x]}{\Longrightarrow} s \text{ and } P \in \{\mathit{QECC},\mathit{Identity}\}\} \end{array}$$

We now prove that $\mathscr{R}$ is a probabilistic branching bisimulation. It suffices to consider transitions between S_i classes, as transitions within classes must be τ and are matched by τ. If $s,t \in S_1(\sigma)$ and $s \xrightarrow{a?[x]} s'$ then $s' \in S_2(\sigma)$ and we find t',t'' such that $t \Longrightarrow t' \xrightarrow{a?[x]} t''$ with $t' \in S_1(\sigma)$ and $t'' \in S_2(\sigma)$, so $(s,t') \in \mathscr{R}$ and $(s',t'') \in \mathscr{R}$ as required. Transitions from $S_2(\sigma)$ are matched similarly. There are no transitions from $S_3(\sigma)$.

There is no need for a probability calculation (case IV of Definition 4) because no probabilistic configurations arise; measurement results are always communicated internally, and never to the external environment.

Finally, because *QECC* and *Identity* have no free variables, their equivalence is trivially preserved by substitutions. □

4 Error Correction: A Second Model

We now consider a different noise model in which random X errors are applied independently to each of the three qubits being transmitted. The new definition of *Noise* is shown below; we use the original definitions of *Alice* and *Bob*; the overall system is now *QECC*2.

$$\begin{array}{l} \mathit{NoiseRnd}(p{:}\widehat{\ }[\mathsf{bit},\mathsf{bit},\mathsf{bit}]) = \\ \quad (\mathsf{qbit}\ u,v,w)\,.\,\{u \mathrel{*}= \mathsf{H}\}\,.\,\{v \mathrel{*}= \mathsf{H}\}\,.\,\{w \mathrel{*}= \mathsf{H}\}\,.\,p![\mathsf{measure}\ u,\mathsf{measure}\ v,\mathsf{measure}\ w]\,.\,\mathbf{0} \\ \mathit{NoiseErr}(b{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}],p{:}\widehat{\ }[\mathsf{bit},\mathsf{bit},\mathsf{bit}],c{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}]) = \\ \quad b?[x{:}\mathsf{Qbit},y{:}\mathsf{Qbit},z{:}\mathsf{Qbit}]\,.\,p?[j{:}\mathsf{bit},k{:}\mathsf{bit},l{:}\mathsf{bit}]\,.\,\{x \mathrel{*}= \mathsf{X}^j\}\,.\,\{y \mathrel{*}= \mathsf{X}^k\}\,.\,\{z \mathrel{*}= \mathsf{X}^l\}\,.\,c![x,y,z]\,.\,\mathbf{0} \\ \mathit{Noise}(b{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}],c{:}\widehat{\ }[\mathsf{Qbit},\mathsf{Qbit},\mathsf{Qbit}]) = (\mathsf{new}\ p)(\mathit{NoiseRnd}(p)\,|\,\mathit{NoiseErr}(b,p,c)) \end{array}$$

$$QECC2(a\!:\!\widehat{\ }[\mathsf{Qbit}], d\!:\!\widehat{\ }[\mathsf{Qbit}]) \quad = \quad (\mathsf{new}\ b,c)(Alice(a,b) \mid Noise(b,c) \mid Bob(c,d))$$

The threefold repetition code is not able to correct multiple errors, so we do not have $QECC2 \cong^c Identity$. The error correction system has a probability of $\frac{1}{2}$ of transmitting a qubit with an X error. We can express this in CQP by using *BitFlip* as a specification process:

$$\begin{array}{rcl} Rnd(p\!:\!\widehat{\ }[\mathsf{bit}]) & = & (\mathsf{qbit}\ u)\{u \mathrel{*}= \mathsf{H}\}\,.\,p![\mathsf{measure}\ u]\,.\,\mathbf{0} \\ Flip(a\!:\!\widehat{\ }[\mathsf{Qbit}], p\!:\!\widehat{\ }[\mathsf{bit}], d\!:\!\widehat{\ }[\mathsf{Qbit}]) & = & a?[x\!:\!\mathsf{Qbit}]\,.\,p?[j\!:\!\mathsf{bit}]\,.\,\{x \mathrel{*}= \mathsf{X}^i\}\,.\,d![x]\,.\,\mathbf{0} \\ BitFlip(a\!:\!\widehat{\ }[\mathsf{Qbit}], d\!:\!\widehat{\ }[\mathsf{Qbit}]) & = & (\mathsf{new}\ p)(Rnd(p) \mid Flip(a,p,d)) \end{array}$$

and by very similar arguments to before, we obtain:

Proposition 2 *$QECC2 \cong^c BitFlip$.*

There is still no probability calculation because the results of the measurements in *NoiseRnd* and *Rnd* are not output. The equal probability of correct and incorrect transmission manifests itself in the fact that the reduced density matrix of the final output qubit, from both *QECC*2 and *BitFlip*, is an equal mixture of the input qubit and its inverse. The only way to introduce probability into this example is for *Flip* to observably output j and *NoiseErr* to observably output the majority value of j,k,l, before the final qubit output.

We know from the standard analysis of this error correction system that if the independent probability of flipping each qubit is $p < \frac{1}{2}$, *QECC*2 reduces the overall probability of a bit-flip error to $p^2(3-2p) < p$. With a slightly more complicated analysis we could also express this property in CQP.

5 Conclusion and Future Work

We have explained the use of the process calculus CQP, and its theory of behavioural equivalence, in analyzing the correctness of quantum communication systems. We have summarized the theory, which is presented in full detail in [2], and given two examples based on a simple quantum error correcting code.

Quantum error correction can easily be analyzed by pen and paper, but the point of process calculus is that it forms part of a systematic methodology for verification of quantum systems. In particular, the congruence property of behavioural equivalence explicitly guarantees that equivalent processes remain equivalent in any context, and supports equational reasoning. For example: we have shown that $QECC \cong^c Identity$; there is a proof in [2] that $Teleport \cong^c Identity$; so we have, for free, that $QECC \cong^c Teleport$, in any context. Because CQP can also express classical behaviour, we have a uniform framework in which to analyze classical and quantum computation and communication.

The next steps for this line of research are to develop equational axiomatizations of behavioural equivalence, in order to reduce the need to explicitly construct bisimulation relations, and to develop software for automatic verification of equivalence. Both of these techniques are already well established for classical process calculus.

References

[1] S. Aaronson & D. Gottesman (2004): *Improved Simulation of Stabilizer Circuits*. *Physical Review A* 70, p. 52328, doi:10.1103/PhysRevA.70.052328.

[2] T. A. S. Davidson (2011): *Formal Verification Techniques using Quantum Process Calculus*. Ph.D. thesis, University of Warwick.

[3] Y. Feng, R. Duan, Z. Ji & M. Ying (2006): *Probabilistic bisimilarities between quantum processes*. Available at `http://www.arxiv.org/abs/cs.LO/0601014`.

[4] Y. Feng, R. Duan & M. Ying (2011): *Bisimulation for quantum processes*. In: *Proceedings of the 38th Annual ACM Symposium on Principles of Programming Languages*, ACM, pp. 523–534, doi:10.1145/1926385.1926446.

[5] S. J. Gay & R. Nagarajan (2005): *Communicating Quantum Processes*. In: *Proceedings of the 32nd Annual ACM Symposium on Principles of Programming Languages*, ACM, pp. 145–157, doi:10.1145/1040305.1040318.

[6] S. J. Gay & R. Nagarajan (2006): *Types and Typechecking for Communicating Quantum Processes*. *Mathematical Structures in Computer Science* 16(3), pp. 375–406, doi:10.1017/S0960129506005263.

[7] S. J. Gay, N. Papanikolaou & R. Nagarajan (2008): *QMC: a model checker for quantum systems*. In: *CAV 2008: Proceedings of the 20th International Conference on Computer Aided Verification*, *Lecture Notes in Computer Science* 5123, Springer, pp. 543–547, doi:10.1007/978-3-540-70545-1_51. Available at `http://arxiv.org/abs/0704.3705`.

[8] S. J. Gay, N. Papanikolaou & R. Nagarajan (2010): *Specification and verification of quantum protocols*. In: *Semantic Techniques in Quantum Computation*, Cambridge University Press, pp. 414–472.

[9] R. J. van Glabbeek & W. P. Weijland (1996): *Branching time and abstraction in bisimulation semantics*. *Journal of the ACM* 43(3), pp. 555–600, doi:10.1145/233551.233556.

[10] M. Lalire (2006): *Relations among quantum processes: bisimilarity and congruence*. *Mathematical Structures in Computer Science* 16(3), pp. 407–428, doi:10.1017/S096012950600524X. Available at `http://arxiv.org/abs/quant-ph/0603274`.

[11] R. Milner (1989): *Communication and Concurrency*. Prentice-Hall.

[12] R. Milner (1999): *Communicating and Mobile Systems: the Pi-Calculus*. Cambridge University Press.

[13] R. Milner, J. Parrow & D. Walker (1992): *A calculus of mobile processes, I*. *Information and Computation* 100(1), pp. 1–40, doi:10.1016/0890-5401(92)90008-4.

[14] M. A. Nielsen & I. L. Chuang (2000): *Quantum Computation and Quantum Information*. Cambridge University Press.

[15] D. Sangiorgi & D. Walker (2001): *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press.

[16] N. Trčka & S. Georgievska (2008): *Branching Bisimulation Congruence for Probabilistic Systems*. *Electronic Notes in Theoretical Computer Science* 220(3), pp. 129 – 143, doi:10.1016/j.entcs.2008.11.023.

[17] M. Ying, Y. Feng, R. Duan & Z. Ji (2009): *An Algebra of Quantum Processes*. *ACM Transactions on Computational Logic* 10(3), pp. 1–36, doi:10.1145/1507244.1507249.