Development and evaluation of a portable raw material mixing system for food extrusion

DJ Kruger
20285841

Dissertation submitted in fulfilment of the requirements for the degree *Master* in Engineering at the Potchefstroom Campus of the North-West University

Supervisor: Prof J Markgraaff

May 2014

Acknowledgements

I would like to thank my heavenly Father for the courage and endurance He has given me to complete this dissertation – without Him, this would not be possible.

I would also like to recognise the contributions that the following people made throughout the duration of this project:

- Casper, Ilna and Liezl-Marié Kruger for their boundless love, support and encouragement.
- Prof. Johan Markgraaff for his guidance, time, patience and reassurance.
- Prof. LJ Grobler for his time and guidance.
- Mr. Piet Van Huyssteen for his guidance regarding the electronic circuitry.
- Bennie Repsold for his assistance in various evaluation procedures.
- Francois Kriel for all the time spent on text editing this dissertation.
- All my friends and colleagues.

Preface

The Centre for Advanced Manufacturing (CFAM) at the North West University's Potchefstroom campus develops extrusion plants. CFAM also assists prospective companies and manufacturers of foodstuffs in undertaking feasibility studies in the development of processed products that would be too expensive to investigate in a full scale plant. Through this, the Centre also provides them with advice on the hardware requirements in their production processes.

An automated pre-processing system is however required where grinding, storing and mixing of ingredients for use during trial runs of food and feed products can be conducted. In addition, experience has shown that in many cases it is more practical to carry out processing trial runs at the existing premises of food producers and that for such purposes, a portable pre-processing facility will aid in these endeavours.

Abstract

In this study, mixing is identified to be the most crucial step during the pre-processing process of extruded food and feed stocks. This study therefore aimed to investigate different mixing techniques in an effort to identify the most effective method and its feasibility to pilot plant application for food extrusion processing. The study furthermore considered the methods of mixing with the view to incorporating the identified method in a standard portable cargo container. The research included an investigation and the design of an inexpensive pre-processing control system that would also save space in such applications where storing facilities for ingredients are housed. After investigating different mixing solutions, a V-blender was identified to be a feasible option. It is suggested that by adding a third leg to the V-blender, to obtain what is dubbed as a “Y”-blender, the effectiveness of mixing would be improved upon - not only in the specified application but with respect to mixing in general.

In order to evaluate and compare the effectiveness of the mixers, rapid prototyping models of a V- and a Y- blender, with capacities of about 7.6 litres each, were produced from medium density fibreboard (MDF) with the aid of a laser cutter. It was found that, for a recipe consisting of 87% fine yellow maize, 12.75% fine sugar and 0.25% colorant, the effectiveness of mixing within the V-blender was greatly influenced by the level to which it was filled. This was not the case for the Y-blender. This therefore suggested that a Y-blender is the ideal solution for the given application.

A layout of a pre-processing system that fits in a standard shipping container and can accommodate six funnel-shaped raw material storage bins with a feed conveyor leading to a Y-blender is designed and a rapid prototyping model of the most vital components of the system is produced. A novel control system using the IOIO USB controller coupled to an Andriod device is developed and this sub-system, with dedicated software, is coupled to the prototyped pre-processing set-up and operated successfully.

Uittreksel

In hierdie studie is die proses van vermenging van bestandele as die mees belangrike stap in die voorverwerkingsproses van ge-ekstrueerde voedsel- en voerprodukte, geïdentifiseer. Die studie was daarop gemik om verskillende mengtegnieke te ondersoek om sodoende die mees effektiewe metode van vermenging te vind en om die haalbaarheid van die graad daarvan, vir die toepassing daarvan op ‘n gids-aanleg vir voedsel-ekstrusie-verwerking, te bepaal. Verder het die studie mengmetodes oorweeg met die uitgangspunt dat die gekose metode in ‘n standaard skeepsvraghouer geïntegreer moet kan word. Die navorsing het ook ‘n ondersoek en ontwerp ingesluit van ‘n goedkoop voorverwerkings-beheerstelsel wat ruimte kan spaar in sulke toepassings waar stoorfasiliteite vir bestanddele, ook gehuisves moet word. Na verskillende mengopsies ondersoek was, word ‘n V-menger geïdentifiseer as ‘n gangbare oplossing. Daar is voorgestel dat die byvoeging van ‘n derde been tot ‘n V-menger, om ‘n sogenaamde Y-menger te verkry, die effektiwiteit van die vermenging sou verbeter – nie alleenlik in die spesifieke toepassing nie, maar ook met betrekking tot effektiewe vermenging van bestanddele oor die algemeen.

Om die effektiwiteit van die mengers te kon vergelyk is snel-prototiperingsmodelle van ‘n V- en ‘n Y-menger, met kapasiteitte van nagenoeg 7,6 liter elk, van medium-digtheid-veselbord (MDF) vervaardig. Daar is gevind dat vir ‘n resep wat uit 87% fyn geel mieliemeel, 12.75% fyn suiker en 0.25% kleurstof bestaan, die effektiwiteit van vermenging van die V-menger grootliks beïnvloed word deur die vlak waartoe die menger gevul word en dat die Y-menger nie tot dieselfde mate deur die vlak van vulling, vir die tipe bestandele wat gebruik is vir die vergelykende toetse, beïnvloed word nie. Daar is vervolgens afgelei dat ‘n Y-menger ‘n ideale oplossing vir die gegewe toepassing is.

‘n Uitleg van ‘n voorverwerkingsstelsel, wat voorsiening maak vir ses tregtervorminge bestandeelstoorbakke en ‘n vervoerband na ‘n Y-menger en as geheel op skaal in ‘n standaard skeepvraghouer kan pas, is ontwerp en ‘n snel-prototiperingsmodel daarvan is vervaardig. ‘n Nuwe goedkoop beheerstelsel wat van ‘n IOIO USB beheerder, gekoppel aan ‘n ‘Android’-toestel gebruik maak, is ontwikkel en hierdie sub-stelsel met die nodige toegewyde sagteware, wat daarvoor geskryf is en met klein veranderinge in praktyk toegepas kan word, is gekoppel aan die vervaardigde model van die voorverwerkings-opstelling en suksesvol getoets en bedryf.

Keywords and definitions

Raw material: Basic recipe ingredients used in food processing.

Mixing: The act or process of blending various ingredients that were initially separated (Smith, 2011).

Extrusion: Raw material, in the form of powder, granules or pellets, is forced through a die to form a product with a constant cross-sectional profile (Wang et al., 2008).

Food extrusion: An extrusion process is used to generate edible products – Raw or cooked products can be produced by this method. For cooked products, an extrusion cooking process is used. The mechanical and thermal energy produced by the extrusion process itself is utilized to cook the food during extrusion cooking (Riaz, 2011).

Extrudate: The products produced by the extrusion process.

Contents

List of figures

List of tables

Chapter 1: Introduction

This chapter provides a short background of the research field. It also provides the problem statement as well the aim of the project.

1.1. Background and problem statement

An extruder is a machine where powder, granules or pellets are fed into a barrel through a hopper. A screw or multiple screws rotating inside the barrel, forces the material through the barrel under elevated temperatures and pressures whilst constantly mixing, shearing and stretching the material. The material is then expelled through a die at the end of the barrel, thus forming products with a constant cross-sectional profile (Wang et al., 2008).

Although extrusion is the process that produces the largest volume of formed plastics according to Kalpakjian & Schmid (2010), Kohlgrüber (2008) stated that it is also used to create an extremely large variety of other shaped raw materials or final products such as in the rubber and food processing industry.

During extrusion cooking in the food processing industry, thermal and mechanical energy is introduced to food and feed ingredients, forcing the basic components of the ingredients, such as starch and protein, to undergo chemical and physical changes (Riaz, 2011).

During food extrusion, there are various factors that influence the quality of the extrudate. These can be factors of the extrusion process itself, or the properties of the material used to produce the products. The factors of the extrusion process include the extrusion temperature, screw speed, feed rate and moisture addition during extrusion and can be altered during the extrusion process to ensure that a high quality extrudate is produced.

Material properties that influence the starch-starch interaction, and therefore also the attributes of the expanded products, include grain size, as well as the oil, salt, protein and sugar content (Mohamed, 1990). These properties cannot be altered by changing the settings on the extruder.

The product recipes consist of ingredients with different particle sizes and densities and the extruder cannot compensate for inconsistent material properties, therefore Sarkar & Wassgren (2009) states that the blending of the recipe ingredients can be seen as the most vital step during the production process. If the recipe is not properly blended or mixed it could increase the difficulty of product handling and can lead to a final product that does not meet the very important quality requirements.

The extruder uses a continuous process, resulting in a need for a continuous supply of raw material, mixed to a fixed recipe, to feed the extruder.

To obtain a reproducible mixture of the recipe, the process requires control of the ingredient addition that also may differ in texture, particle size, moisture content and prior compaction.

In general mixing systems are readily available in the industry, but these systems are expensive, normally fixed and typically occupy relatively large areas to deliver a maximum throughput. Due to the large size of these systems, the development of new recipes or the fine-tuning of existing recipes are impeded due to the high cost associated with large scale experimentation.

A need therefore exists to develop a portable mixing system that would provide an opportunity for producers to develop new recipes on a small scale with a system that can be added to an existing plant or by the use of such a system housed in a container, transported to development sites.

To date there is no standardised method for selecting and incorporating appropriate mixing systems based on parameters such as particle size, density, etc. Therefore, a study is required to evaluate the mixing process and various existing mixing systems.

1.2. General aim

Mixing plays an integral part in pre-processing of food and feedstock extrusion processes. This study aims to review available mixing methods in order to identify the most effective method of mixing with the view to incorporating such method into a portable pre-processing plant and obtaining a homogeneous mix. It is a further aim of this research to design a pre-processing system that includes the required vital components inclusive of an accurate and inexpensive control system.

It is required that the pre-processing and the associated control system fit into a standard shipping container (Dimensions: L = 5900mm; H = 2400mm; W = 2350mm), and that subsystems be incorporated or selected to deliver a minimum of 350 kg/h of selected recipes.

The system must be capable of manipulating six different materials simultaneously. These materials may include:

- Super maize meal (Density: 700 - 750kg/m^3)
- De-fatted soy flour (Density: 600 - 650kg/m^3)
- Wheat flour (Density: 750 - 800kg/m^3)
- Rolled oats (Density: 300 - 450kg/m^3)
- Wheat germ (Density: 650 - 700kg/m^3)
- Sugar (Density: 800 - 950kg/m^3)

The mixing system forms the main focus of this study. It is therefore stated that the material of the recipes is provided to the system in such a manner that no grinding or similar pre-processing other than mixing is required. Other ingredients such as salt, flavourings, vitamin mixtures and sunflower oil, if required, are to be added manually to the mixer.

To be practical, the design is limited to providing for milled ingredients in order to eliminate space occupation by grinding facilities since the ingredients listed are usually available in milled form. The design therefore only considered incorporation of storing, mixing, and weighed feeding for direct or immediate extrusion processing.

Chapter 2: Literature review

Chapter 1 provides the reader with a short background pertaining to the extrusion process and identifies mixing as a critical step during the manufacture of an extrudate. This chapter provides a literature study covering factors which influence the mixing process, as well as commercial mixing systems with their alternatives. Different mixer alternatives are evaluated according to their properties and a suitable mixing system is suggested that also leads to the definition of the study scope.

2.1. Mixing concepts

According to Ottino & Khakhar (2000) there exist mainly three modes of mixing: shaking granular material, tumbling the material or driving impellers or paddles of some sort through the material. They primarily studied the effect of tumbling and identified container shape, the degree of filling, the mechanism used to induce particle dispersion, differences in material properties and particle interactions as the properties that most affect mixing. These properties are discussed individually in the following section of the study.

There are two different types of mixers: continuous and batch mixers. These mixers can also be separated into three distinct classes namely: gravity driven-, stirred- and high shear–mixers (Cleary & Sinnott, 2008).

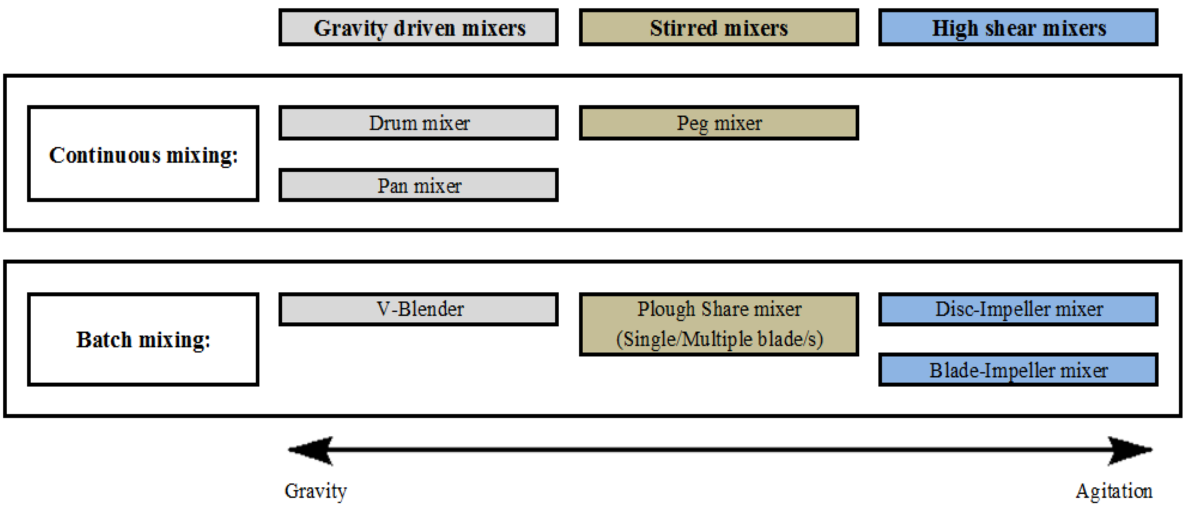

Figure 1: Mixer types arranged according to mixing class or method of mixing (gravity driven/stirred/high-shear) that can be configured for continuous or batch mixing (modified after Cleary & Sinnot, 2008).

Figure 1, as adapted from Cleary & Sinnott (2008), illustrates the typical relation between mixer types and the method of mixing employed.

A batch mixer is a device that receives a full set of material to blend. The device will then mix the material after which the full mixture is removed before the next set of un-blended material is introduced. A continuous mixer will continuously receive a blend of unmixed material and mix the material as it passes through the device. The blended material is extracted from the mixer as new material is introduced, making it a continuous process (Cleary & Sinnott, 2008).

The difference between gravity driven, stirred and high shear mixers is that a gravity driven mixer is a device that utilizes the earth's gravitational force by passing material over a rotating or tumbling surface to mix or blend the material. If a mechanical agitator, like a peg or a blade, is used to induce particle dispersion, the device is classified as a stirred mixer except for the special class of stirred mixers where a very rapid moving agitator is used. This special class is then identified as a high shear mixer.

Two distinct techniques exist that should be considered when optimizing a mixing process. The first technique is where a mixer is required for creating a limited variety of mixtures with long production runs, as opposed to mixers producing a large variety of mixtures with shorter production runs. Finding a single mixer with favourable attributes for both scenarios can prove difficult, but can be done if the properties that influence material dispersion are recognized and controlled in an optimal manner (Borzenski, 1999).

Various mixing directions are defined in Figure 2, which will be used to discuss mixing characteristics and these directions will be referred to throughout the rest of the text.

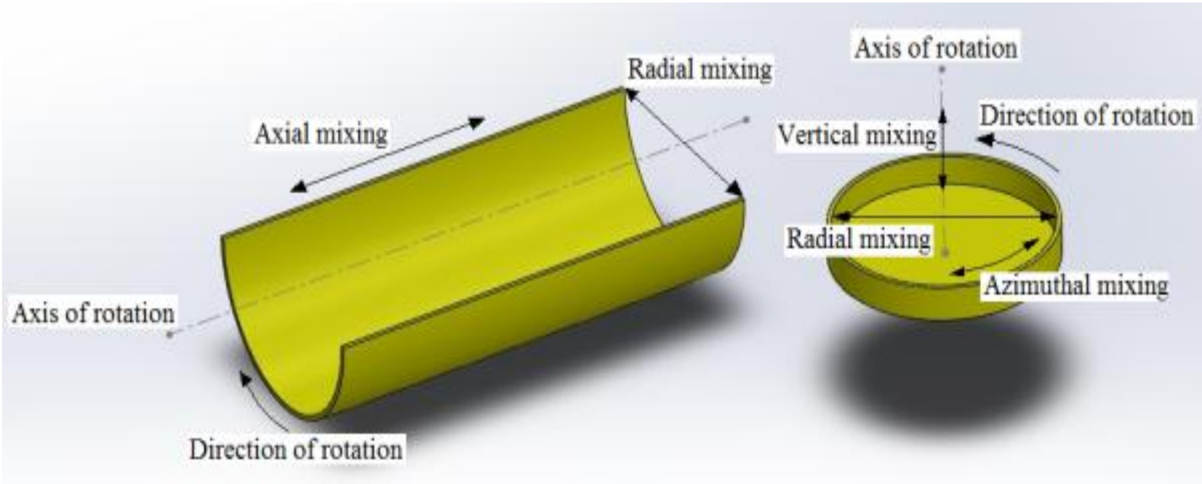

Figure 2: Definition of mixing directions used to describe the mixing processes throughout the rest of the text.

2.2. Evaluation of mixing effectiveness

According to Smith (2011) it is important to define the difference between mixing and pure agitation, where agitation is the action of inducing a particular pattern of the material flow, whereas mixing is defined as the act of diffusing various ingredients that were initially separated. Mixing can therefore be caused by agitation. Smith (2011) identified three characteristics by which the effectiveness of mixing can be assessed namely:

- ➔ The degree to which the ingredients mixed (*level of mix*)
- ➔ Required time to obtain the level of mix (*mix time*)
- ➔ Power consumption rate

Each of these characteristics will now be discussed individually.

Level of mix

The *level of mix* represents the degree of uniformity throughout the blend of ingredients. Spot samples are taken from the mixer and the concentrations of components within these samples are then compared to average or expected values. The material is completely mixed when one component is randomly distributed within another. A variance or standard deviation can be used to compare the composition of the sample taken to the expected or an average value. The equation

$$s^2 = \sum \frac{(x_i - \bar{x})}{n-1},$$

defines the variance s^2, where x_i represents the specific local value, $\bar{x}$ represents the expected or average value and *n* is the quantity of samples taken. The variance s^2 of unmixed material is given by the equation

$$s_0^2 = p(1-p),$$

where the ratio of a component is given by p. Because s_0^2 is only obtainable in theory an acceptable variance of s_∞^2 is defined as a limiting value to lean toward. The value of s_∞^2 will therefore not be exactly 0, but will tend towards it.

Smith (2011) defines *point uniformity* as the ideal uniformity of the mixture, which is especially not feasible for the mixing of solids, and *overall uniformity* where mixing occurred until an acceptable level of variance was reached. Smith (2011) explained the difference between *point uniformity* and *overall uniformity*, by making use of illustrations of the difference in distribution of black and white blocks as shown in Figure 3 and Figure 4, respectively.

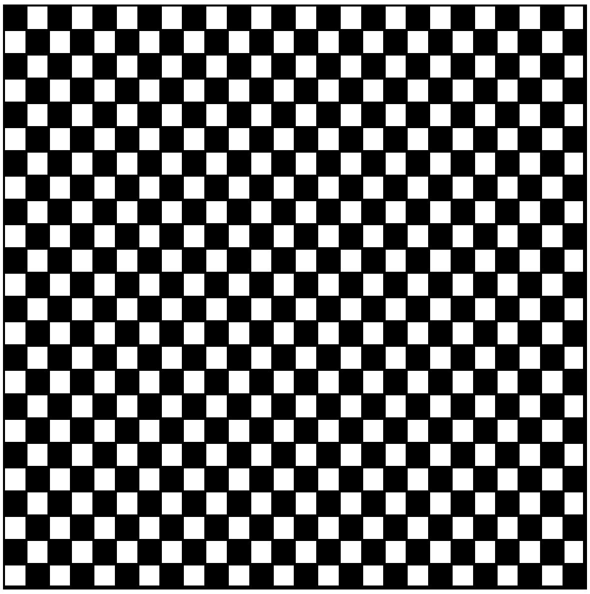

Figure 3: Point uniformity.

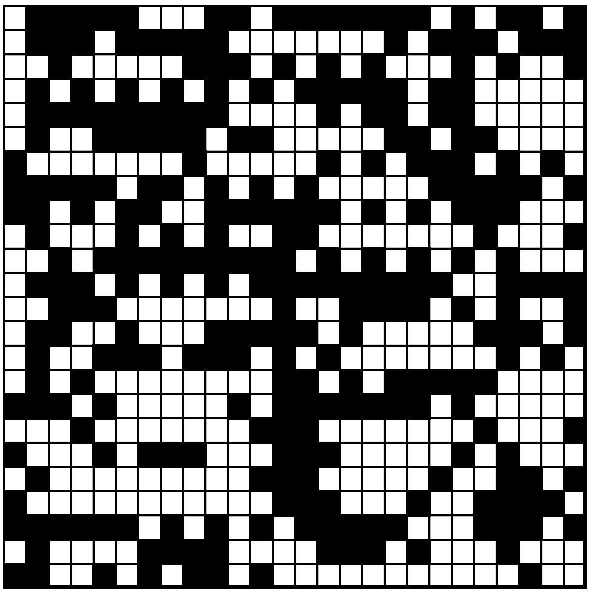

Figure 4: Overall uniformity.

A similar approach to define the *level of mixing* within a sample was presented by Aissa *et al.* (2011), where they evaluated the effect of density and of friction coefficient on the mixing of linear medium density polyethylene powders, polyetherimide powders, glass beads and wood powder. They evaluated each type of material individually, varying only the colour of the material to be mixed.

They applied a Grey Level Co-occurrence Method (GLCM) to experimentally evaluate the mixing degree. It is a process where a camera is positioned above the mixture and a photo is taken of the mixture. The pixels of the image are then converted to greyscale. A co-occurrence matrix is then created containing the relationships of neighbouring pixels across the image.

Similar to the variance (s^2), as defined earlier, an *area fraction* is then determined in order to evaluate the uniformity of the mixture composition, where:

$$A_f = \frac{N_C}{N_T}$$

with:

A_f = Area fraction of the selected colour within the image

N_C = Number of pixels of the selected colour within the image

N_T = Total number of pixels within the image (entire range of colours)

Mix time

According to Smith (2011) the scale of scrutiny is dependent on the size of the sample used to evaluate the mixture. Smith (2011) identified two methods to define the scale of scrutiny for a specific case.

The minimum size of the sample taken must be the sample size required to cause the mixture to be considered an imperfect mixture. The size and/or volume of the end-product therefore define the size and/or the volume of the sample to be observed.

It is known that there isn't a direct correlation between variance and time; a dimensionless fractional measure is required to correlate with time and therefore, a mixing index (M) was introduced by Smith (2011). This dimensionless measure (M) can be obtained by various methods of calculation where the most common are (Smith, 2011):

$$M = \frac{s^2 - s_\infty^2}{s_0^2 - s_\infty^2} \qquad (1)$$

$$M = \frac{s - s_\infty}{s_0 - s_\infty} \qquad (2)$$

$$M = \frac{\ln s - \ln s_\infty}{\ln s_0 - \ln s_\infty} \qquad (3)$$

$$M = \frac{s_0^2 - s^2}{s_0^2 - s_\infty^2} \qquad (4)$$

$$M = \frac{s_0 - s}{s_0 - s_\infty} \qquad (5)$$

If equation (1) is used, the difference between the variance at a specific point in time (*t*) and the variance at the end state, can be defined as $s^2 - s_\infty^2$. If the rate at which mixing is taking place is represented by a constant *k*, the rate of change in variance is given by

$$\frac{d(s^2)}{dt} = -k(s^2 - s_\infty^2).$$

Thus, after integration between s_0^2 at $t = 0$ and s^2 at time t and substitution into the first equation, the Mixing Index (*M*) is obtained, where:

$$M = e^{-kt}.$$

The experimental values of the algorithm can now be plotted against time and extrapolated to determine the desired level of *M* and the required mix time (Smith, 2011).

Power consumption rate

The method of calculating power requirements varies with different mixing or blending devices and no overall technique to evaluate power consumption rate therefore exist. For each device under consideration a tailored calculation method must be used.

2.3. Simulation and verification of mixing characteristics

The Discrete Element Method (DEM) is used to create simulations of the motion of particles within a mixer. According to Metcalfe *et al*. (1998), each particle is assigned a spring constant (k), a dashpot damping coefficient (C), a coefficient of friction (μ) and the maximum allowable particle overlap (Δx) is specified. A simplified representation of such a contact force model is shown in Figure 5 as adapted from Metcalfe *et al.* (1998). In the figure the suggested relationship between two of the particles is given. The same procedure is followed for the interactions between all the touching particles within the mixer and the interactions between the particles and the mixer/container wall is also specified. These properties are then used in computer simulations to predict the influence of the mixing device design on the particle mixing, as well as the influence of the particles on each other. This method provides researchers with a procedure to predict the behaviour of various mixers to a certain degree of accuracy.

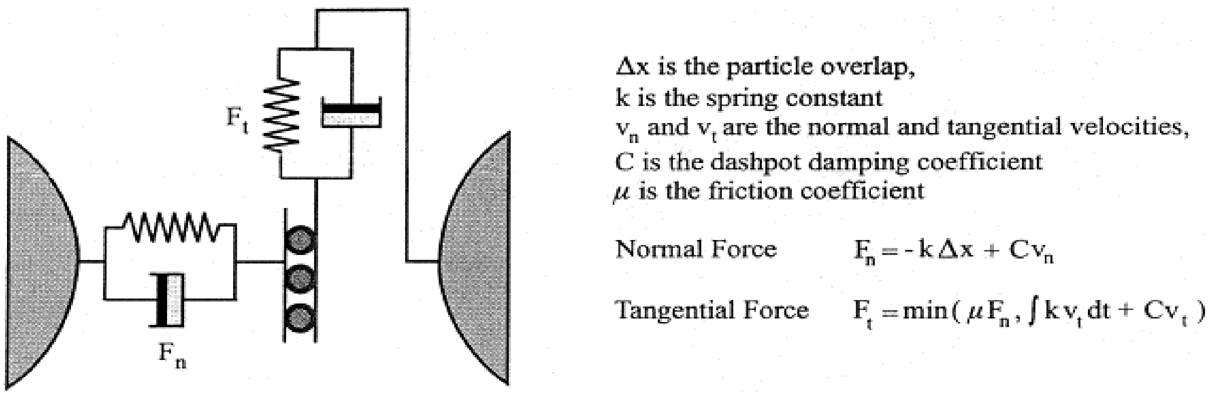

Figure 5: Contact force model (Metcalfe et al., 1998).

Positron Emission Particle Tracking (PEPT) can be used to evaluate the behaviour of an actual mixer and therefore to evaluate the validity of DEM simulations. Stewart *et al.* (2001) stated that PEPT was founded on the principle of locating gamma rays produced from positron decay caused by a trace element added to the particles being mixed. Two gamma ray detectors – one on each side of the mixer – make up a positron camera. When both of the plates of the camera detect a gamma ray at the same time, the tracer element must lie on a line connecting these two points. The locations of the tracer element can then be plotted over time on a three-dimensional chart to emulate the movement of a particle within the mixer. This data can then be compared to DEM simulation to verify the modelling.

2.4. Factors affecting mixing

The first aspect to be assessed is the geometry of the container. Khakhar *et al*. (1999) simulated tumbling mixers by computation of *Poincaré sections* as well as by *blob deformation* as explained below.

To define the *Poincaré sections*, several initial positions within the particle bed were selected and the particle trajectories were then computed by integrating the velocity fields of the particles with respect to time. These sections are then used to represent streamlines or flow patterns that the particles follow as the mixers tumble.

For the *blob deformation* simulations, groups of particles were selected inside circular regions within the particle bed. As the mixer rotated, the positions of all of the selected particles were mapped and the velocity fields of the selected particles were integrated over the specified time duration. This also produced streamlines or flow patterns for the observed particles.

Figure 6 shows the results of the simulations for round, elliptical and square mixing containers, as established by Khakhar *et al.* (1999). The images in the top left corners of Part A, B and C represent the streamlines or flow patterns produced by the Poincaré sections. The images in the top right corners of Part A, B and C represent the initial positions of two selected groups of particles used to simulate the blob deformation. These groups of particles were coloured red and blue respectively. The images at the bottom of Part A, B and C represent the streamlines or flow patterns produced by the blob deformation simulations.

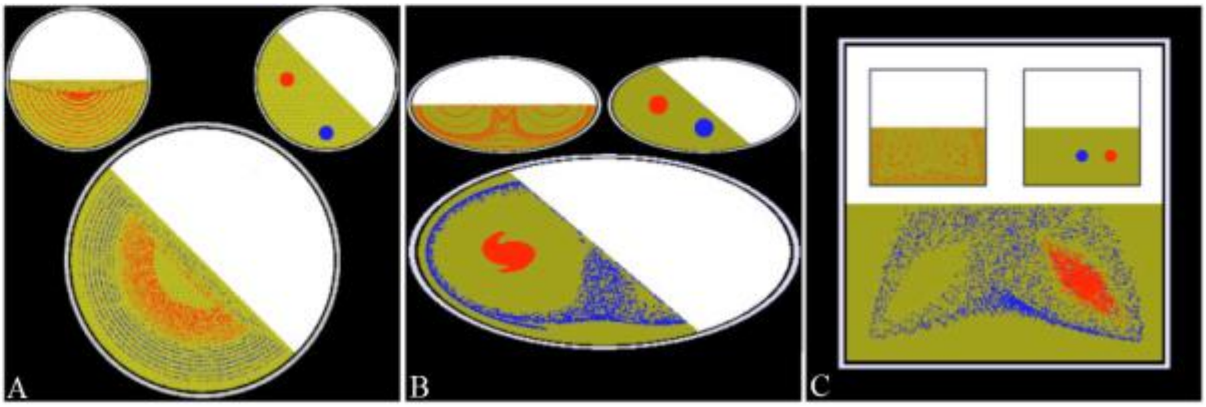

Figure 6: The influence of the cross-sectional profile of the mixer on the flow behaviour of particles over time as adapted from Khakhar *et al.* (1999). A full description of the images is provided in the text.

In Part A of Figure 6, the flow patterns of the particles give rise to ordered or near elliptical streamlines. As the aspect ratio of the container is changed, as can be seen in Part B of the Figure 6, the flow patterns become distorted in the blue areas, giving rise to more chaotic regions. These chaotic areas imply that the streamlines of the various particles will eventually intersect each other. In Part C of Figure 6 a square container is used and a further increase in the size of the chaotic regions is noted. This motion of mixing by inducing chaotic flow patterns within the mixer is known as chaotic advection.

According to Ottino & Khakhar (2000) chaotic advection has always been fundamental in the science of the mixing of liquids and granular powders. They stated that varying the streamlines of the flowing particles over time is enough to induce chaotic advection within the material and that it is this chaotic motion that promotes better mixing as opposed to regular regions where the particle flow is ordered and poor mixing usually arises.

Through the data presented by Khakhar *et al.* (1999) and Ottino & Khakhar (2000), it was found that the simplest method to induce chaotic advection, and therefore promote better mixing in tumbling systems,was to change the mixer geometry to a non-circular cross-section. Part B in Figure 6 illustrates how a change in the aspect ratio of a circular mixer produced this effect. Kuo *et al*. (2002) used a V-shaped mixer to induce this principle of utilising drum shape for improved mixing.

The second aspect to be assessed is the flow regimes that the particles are subjected to within a mixer.

According to Ingram *et al*. (2005) a bed of material within a rotating drum consists of an active and a passive layer. The passive layer is the bottom layer and rotates in accordance with the drum at fixed radius and axial position. In this passive layer a plug-flow is formed and there is little interaction between the particles in this region.

The upper layer of material forms the active layer where the particles are less compact and can move in relation to each other. It is therefore safe to assume that, for tumbling mixers, the majority of the mixing takes place within this layer.

Between the active and the passive layer is an interface, commonly referred to as the *yield line*. As the mixing container rotates, the friction between the material and the container wall will cause the material to form a slope in the direction of rotation. At low rotational speeds a slope of material will build up until a certain angle $\beta_{i,}$ at which the gravitational pull is large enough to cause the upper layer of the material to start sliding down to the base of the container, to form a slope with a new angle β_f (Ottino & Khakhar, 2000).

The process will continue to occur as long as the drum is rotating causing an avalanching effect. As the rotation speed of the mixer increases, a curved bed will form on the slope and the avalanching effect would become more continuous as particles circulate from the bottom of the sliding layer to the top causing a rolling, cascading effect.

If the speed of rotation is further increased, the centrifugal force will increase, causing the particles to be raised higher along the container wall. As the centrifugal force is overcome the particles move away from the container wall and follow an arc back to the bed of materials below. This movement is described by Finnie *et al.* (2005) as the cataracting motion.

A further increase in the speed of rotation of the mixer will initiate a centrifuging motion. This implies that the centrifugal force is great enough to overcome the gravitational pull on the particles for the full rotation of the drum. This motion will cause all the particles to be confined to the passive layer and therefore very little, if any, mixing will take place other than mixing caused by segregation (Segregation is the occurrence where material is separated naturally according to particle size and density).

The most desirable mixing mechanism is the cataracting motion, with the centrifuging motion being the least desirable. Figure 7 shows the active and passive layers within the flow regime of the particles as adapted from Ingram *et al.* (2005), whereas Figure 8, as adapted from Ottino & Khakhar

(2000), shows various mixing mechanisms that can be found in tumbling mixers as discussed above. The symbol 'θ' in Figure 7 represents the angle between the horizontal and the active trajectory, whereas X represents the direction of particle flow over the active trajectory.

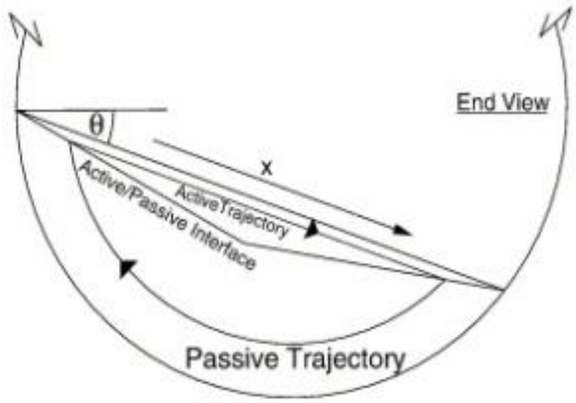

Figure 7: Active and passive layers within the flow regime of the material particles as described within the text (Ingram et al., 2005).

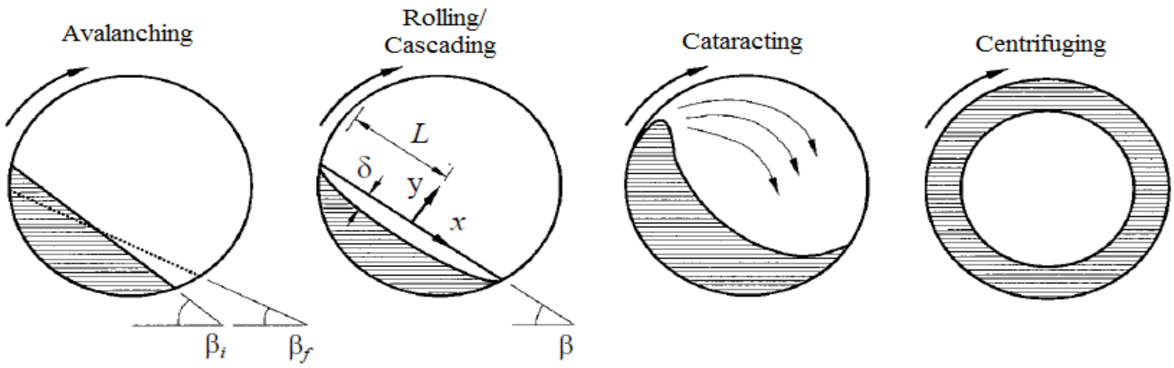

Figure 8: Various mixing mechanisms as found in tumbling mixers as adapted from Ottino & Khakhar (2000). The definitions of β, β_i, and β_f are discussed in the text.

The degree to which a mixer is filled is another element of mixing efficiency. Remy *et al.* (2010) states that the level to which a mixer is filled is critical to the effectiveness of the mixer in agitated devices or bladed mixers and Ottino & Khakhar (2000) confirms that this is also the case in tumbling barrels where the degree of external agitation is at a minimum.

Sarkar & Wassgren (2009) as well as Sarkar & Wassgren (2010) used Discrete Element Method (DEM) simulations (DEM is explained in Section 2.3) to imitate the working of an agitated, continuous, horizontal drum mixer. To correctly compare the mixing properties for varying mixer configurations, a dimensionless form of reporting the rotational speed was introduced by the Froude number (Fr).

The Froude number represents the correlation of centripetal acceleration over the tips of the blades in a bladed mixer to the acceleration caused by gravity. The Froude number is calculated by:

$$Fr = \frac{\omega^2 \times D_{mixer}}{2 \times g}$$

Here ω is the rotational blade speed, D_{mixer} is the diameter of the drum and g is the acceleration due to gravity.

They found that the granular material bed became fluidized at small fills with large impeller speeds. This fluidized regime resulted in the best mixing per blender shaft rotation as this configuration produced significant particle dispersion and high axial flow rates. They also stated that higher fills (more than two thirds of the mixer volume) hinders the mixing process as the lack of free space within the mixer, as well as the increased cohesion values between the particles, limits the movement of the particles in the radial direction.

Laurent & Bridgewater (2002) also state that, with increasing fill level, the rotational speed of the particle bed will increase until it approximates the rotational speed of the mixer blades. This emulates the centrifuging mixing mechanism (as previously discussed), which leads to a decrease in the level of mixing (the centrifuging mechanism fully occur at Fr = 1).

It was found that the fill level influenced the residence time within the continuous mixer and, as increased residence time has different influences on the degree of mixing in the different mixing directions, the influence of the fill level also had different influences on the degree of mixing in the different mixing directions.

For the investigated horizontal drum mixer discussed above, Sarkar & Wassgren (2010) found the highest level of mixing in the radial direction at a fill level of approximately 55% and the highest level of mixing in the axial direction at a fill level of approximately 25%.

Part A of Figure 9 shows the influence of increasing Froude number on a horizontal blade mixer filled to 31% of its volume, whereas Part B shows the influence of increasing Froude number on this type of mixer filled to 63% of its volume. In Figure 9 the blue particles are the lowest velocity particles as opposed to the red, highest velocity particles.

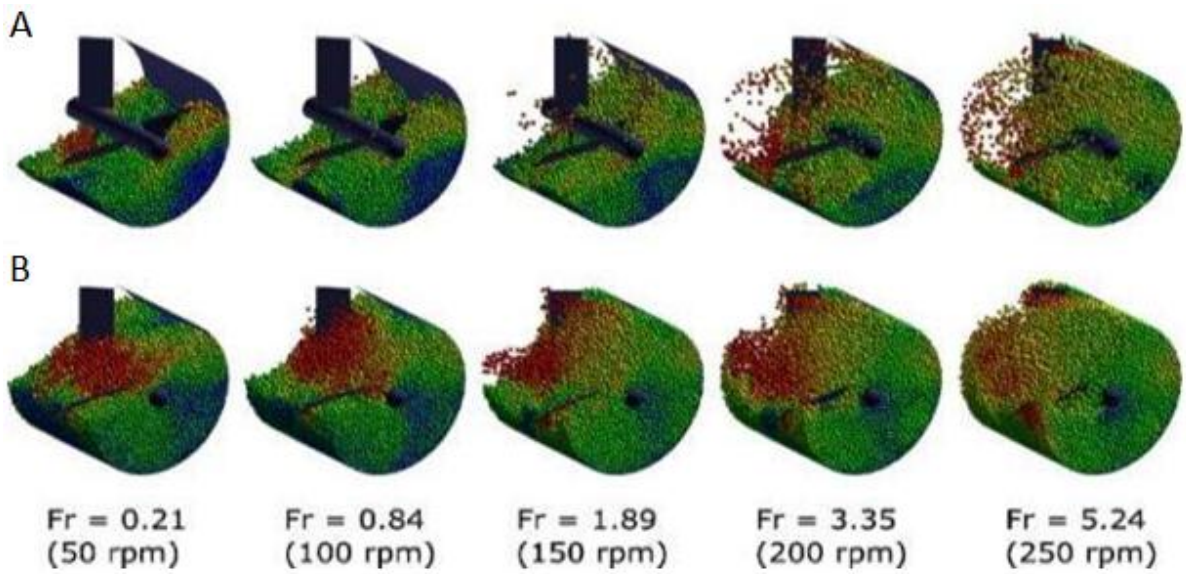

Figure 9: The influence of Froude number on particle velocities and particle flow (Red = highest velocity particles, Blue = lowest velocity particles). Part A shows the mixer at a fill level of 31% and Part B shows the mixer at a fill level of 63% (Sarkar & Wassgren, 2010).

Through the results of Figure 9, it was establish that an increase in Fr at a high fill level causes the bulk of the material within the mixer to flow as a solid mass at a constant speed. This action causes very little mixing to occur and therefore is unwanted. Figure 9 also shows that for a lower fill level, even with a high Fr value, the individual particle velocities still vary throughout the particle bed. This suggests that higher mixing rates are possible at lower fill levels.

Remy *et al.* (2010) investigated a vertical drum mixer with a rotating blade at the bottom of the drum as shown in Figure 10. They defined an H/D value to represent the fill level of the drum, where H is the height of the particle bed prior to blade rotation and D represents the inner diameter of the drum. An H/D of 0.17 was given as the fill level required for the particle bed to only just cover the height of the blades.

The fill level was incrementally increased and conclusions were drawn on the effect thereof. They found that, at a fill level of H/D = 0.17, a heap was produced in front each of the blades while valleys formed behind them. This motion was identified to promote both radial and vertical mixing. They found that, although this phenomenon was found for all fill levels, the magnitude of it varied with variation in fill level.

Mixing in the azimuthal direction was mostly a function of wall friction and the rotational speed of the impeller. It was found that the greatest degree of mixing in the vertical and radial direction was found for a fill level that was just able to cover the height of the blades (H/D = 0.17). A further increase in fill level for this type of mixer produced a decline in the degree of mixing.

Figure 11 shows the influence of H/D on mixing. In the figure the mixer was filled to three different levels (H/D = 0.17, H/D = 0.32 and H/D =0.46) with particles defined by their initial position (The portion of particles on the left was coloured yellow and the portion of particles on the right was coloured red). The figure first shows the particle dispersions at three time periods at the top of the particle bed in Part A and then shows the particle dispersions for all three levels of fill at a section view through the particle bed just above the blades in Part B.

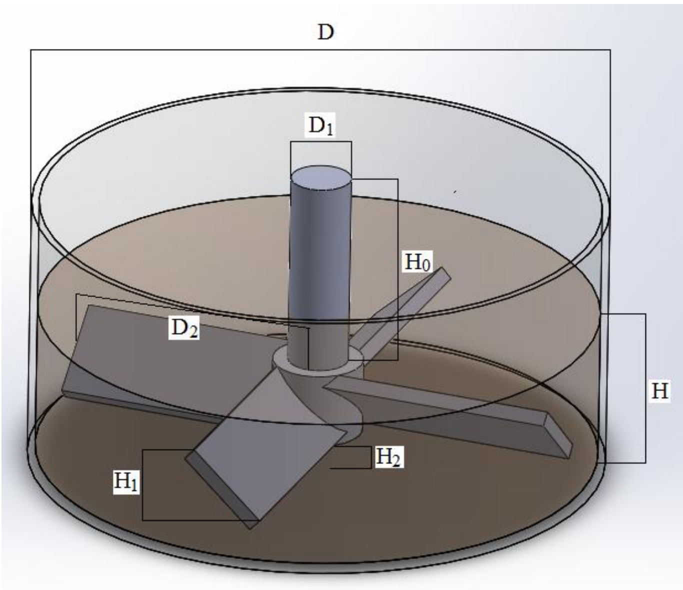

Figure 10: Schematic of vertical drum mixer giving H as the height of the particle bed within the mixer and D as the diameter of the mixer.

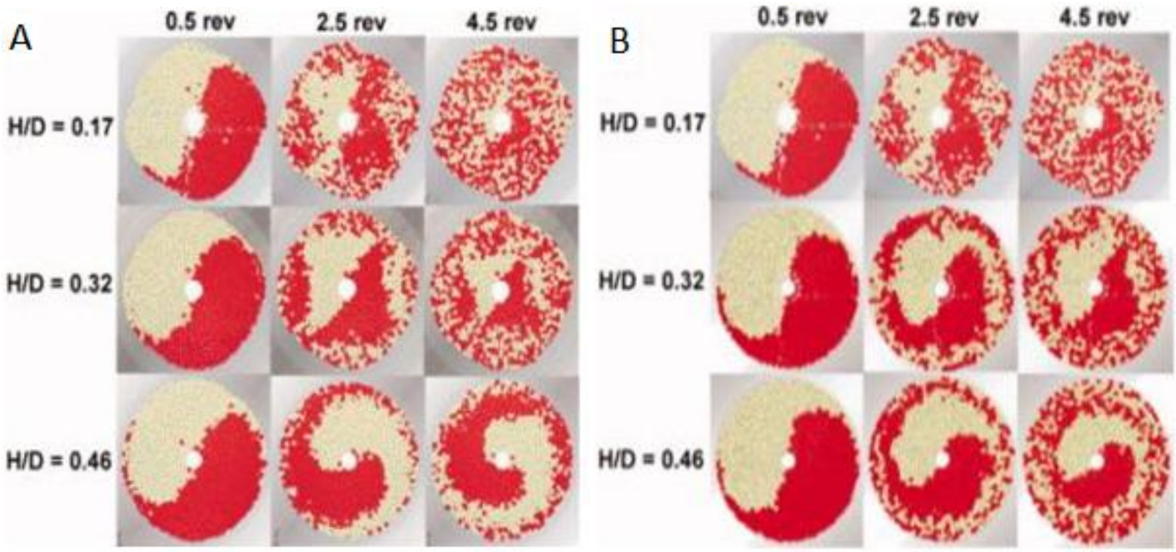

Figure 11: The influence of H/D on the degree of mixing. Part A shows a top view of the particle bed and Part B shows a section view of the particle bed as found at blade height: H_1+H_2, as defined in Figure 10.

After examining the results found in Figure 11, it was found that, for a H/D ratio of 0.17, uniform blend uniformity was reached after 4.5 revolutions. As the fill level was increased from there, the uniformity of the mixture after 4.5 revolutions was reduced. By comparing the results of Part A and Part B of the figure, it can be seen that, even for the higher H/D ratios, the majority of the mixing only took place from the bottom of the particle bed up to the height of the mixer blades. It is therefore determined that any increase in fill level higher than the mixer blades only hinders the mixing process for any number of blade revolutions for this type of mixer.

Cleary & Sinnott (2008) evaluated various mixers and mixing characteristics using DEM Simulations and they were able to correlate their findings with other accepted studies. To evaluate the effect of particle shape on the level of mixing in a bladed-impeller high shear mixer, they compared simulations of spherical particles and of box shaped particles with rounded edges which they labelled as Super Quadratic (SQ) particles.

The SQ particles are defined to occupy the same volume per particle as a spherical particle. They found that, due to the greater contact area between the SQ particles caused by the longer and flatter surface areas, the particle bed's resistance to shear was greatly increased. This in turn caused the force required to agitate the particle bed to increase.

Cleary & Sinnott (2008) found that, for ten revolutions in the mixer, the SQ particles were mixed 14% less effectively in the radial direction as compared to the spherical particles. The effectiveness of the mixing of the SQ particles was increased by 3% in the vertical direction, whilst the mixing effectiveness in the azimuthal direction decreased by 4%.

Through these results they were able to conclude that particle shape had an undeniable effect on the degree of mixing and that the effect varies in the different mixing directions. Table 1 shows the mixing effectiveness that they obtained when mixing the different shaped particles for ten revolutions in the bladed-impeller high shear mixer described above.

Table 1: Mixing effectiveness in the different mixing directions for particles of different shape, as adapted from Cleary & Sinnott (2008).

Particle shape	Mixing direction		
	Azimuthal	Radial	Vertical
Spherical	87%	69%	73%
Super-quadratic (SQ)	83%	55%	76%

Aissa *et al.* (2011) proved through the use of the experimental Grey Level Co-occurrence Method (GLCM) that density, size and other particle variations are directly linked to the threat of flow induced material segregation and therefore mixture homogeneity (GLCM is described in Section 2.2).

According to Ottino & Khakhar (2000) this is a trend that cannot be found in the mixing of liquids, which makes it a problem unique to particle mixing. They stated that segregation occurs when flow is induced over particles with varying size and density, which causes the particles to separate according to either their size, density or both.

They stated that most models created thus far either simulate segregation where the particle sizes are identical and the mass differs, or where the particle sizes differ and the mass are the same.

It is therefore difficult to accurately simulate particle behaviour for materials with both particle size and mass variations and product recipes should therefore be designed and selected so that at least one of these variables are similar for all the materials to be mixed.

Due to the fact that this is not always possible, a buffer material is often required to either bridge the gap between the variations in the material properties or to increase the viscosity of the flow medium of the material to be mixed.

2.5. Study of typical commercial mixers

The previous sections discussed the various mixing concepts and the factors affecting mixing. This section will now discuss some of the typical commercially available mixers based on the principles covered by the previous sections.

Drum mixer:

A drum mixer consists of a rotating drum where material is introduced at the one side of the drum and expelled from the other. The mixer can be inclined to ensure the direction of material flow, and the size of the outlet can be reduced to increase the material retention time.

Cleary & Sinnott (2008) utilized DEM models to simulate and investigate various mixers. They state that drum mixers can be effectively operated as continuous or batch units, it was however only simulated as a continuous unit.

During the simulation it was found that, as particles entered the mixer they immediately started rotating in a rolling regime, as discussed earlier. It was found that, as smaller particles moved through the active layer, they pass through the voids in the passive layer. Due to this, the larger particles were forced out of the centre towards the active layers, thus inducing size segregation within the drum mixer, which is an unattractive attribute for mixers.

Another drawback with the rotating mixer is that all the material is transported axially towards the exit as a uniform plug, with minimal axial mixing taking place during the process. These results indicated that this unit does not perform well as a mixing device.

Figure 12 shows the formation of plug flow, where the mixer is first filled with blue coloured particles. As red particles are introduced, two distinct particle flows can be seen with very little axial mixing taking place over the particle beds. A thin layer of blue particles can also be seen forming a stagnant particle layer at the entrance of the mixer.

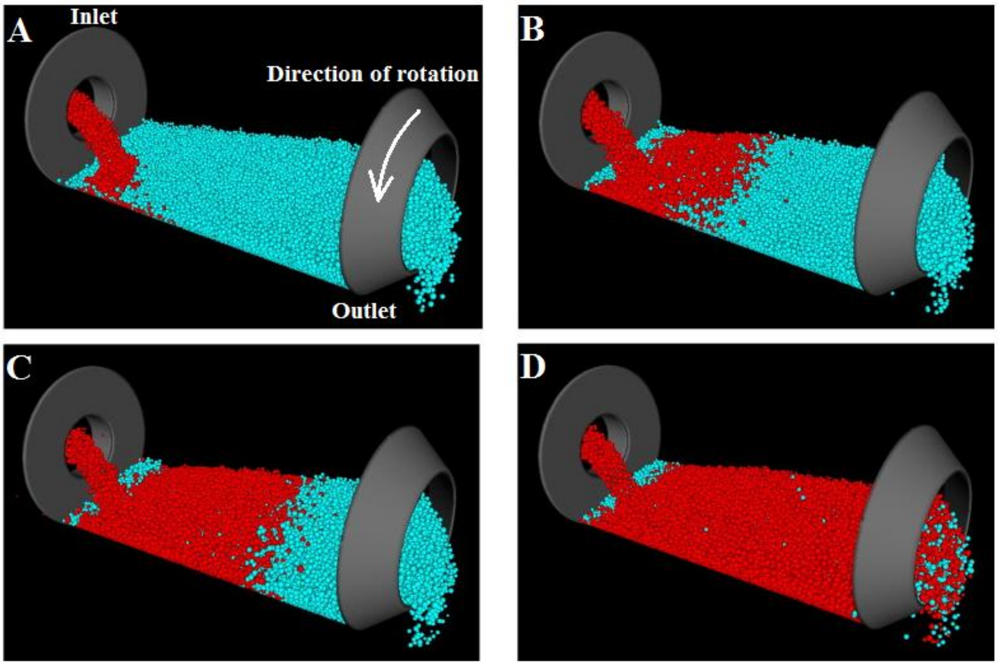

Figure 12: The formation of plug-flow within a continuous drum mixer. In Part A material with the same properties, but different colour, starts to be fed into the mixer. In Parts B and C, it can be seen how the blue particles are being forced forward, with little mixing occurring between the two sets of material.

Pan mixer:

In another simulation by Cleary & Sinnott (2008) a pan mixer was studied. A shallow, flat bottomed pan rotating at an angle is the principle of operation of this type of mixer. Material was poured onto the upper side of the pan where after it followed swirling trajectories over the pan before reaching a bed forming at the lowered edge of the pan. Once enough material was introduced, the wall overflowed, discharging the material.

Initially high particle velocities were observed during the simulation, but all the particles followed the same path as they were introduced. This also gave rise to a plug-flow as in the case with the drum mixer, which in turn results in poor mixing.

Figure 13 shows the particle speeds over the mixer, with the red particles being those with the highest particle speeds and the blue particles being the particles with the slowest particle speeds. As can be seen from the figure, all the particles enter the mixer at high speed and, as they move across the mixer, they slow down. The particles then enter a near stationary state at the pan wall until they are forced over the wall by the subsequent particles, generating a plug flow principle.

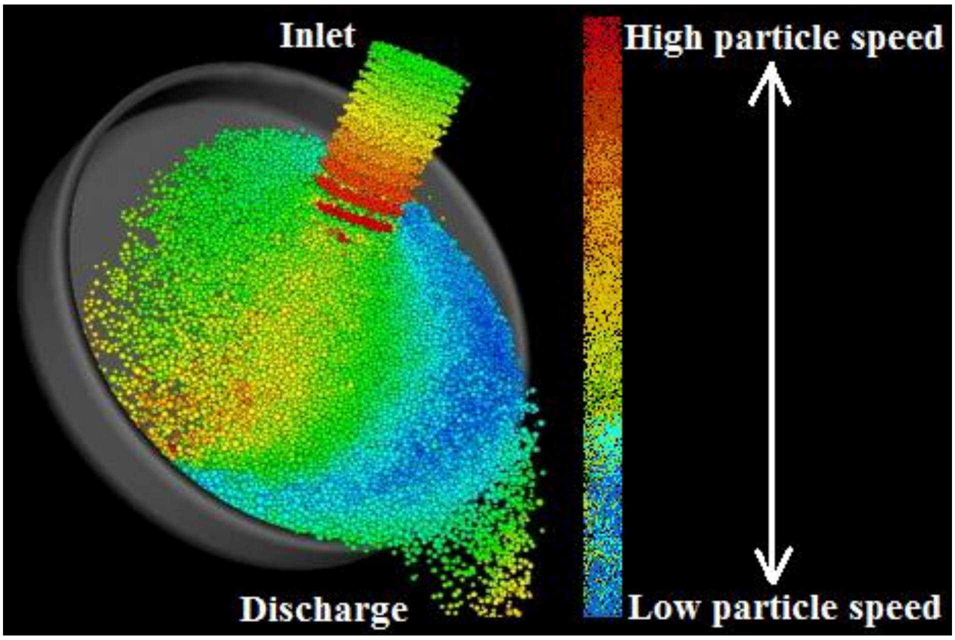

Figure 13: Formation of plug flow within an inclined pan mixer. This is indicated by high particle speeds as particles enter the mixer and the lower particle speeds at the wall of the mixer just prior to discharge.

V-blender:

The V-blender consists of two cylindrical containers joined at one point to create a single V-shaped container. The container is rotated so that the intersection of the two legs faces upwards (^). The unit is filled with material through an opening at this intersection, before this end is sealed and the unit is tumbled. After mixing, the material is extracted from the same opening.

Kuo *et al.* (2002) created various DEM simulations of such a V-blender and verified their results with Positron Emission Particle Tracking (PEPT is discussed in Section 2.3). It was found that although good mixing occurs within each of the separate legs, the mixing across the boundary between the legs is very poor.

Figure 14 shows how a good level of mixing is achieved within each leg of the mixer, but poor mixing is achieved over the boundary between the legs.

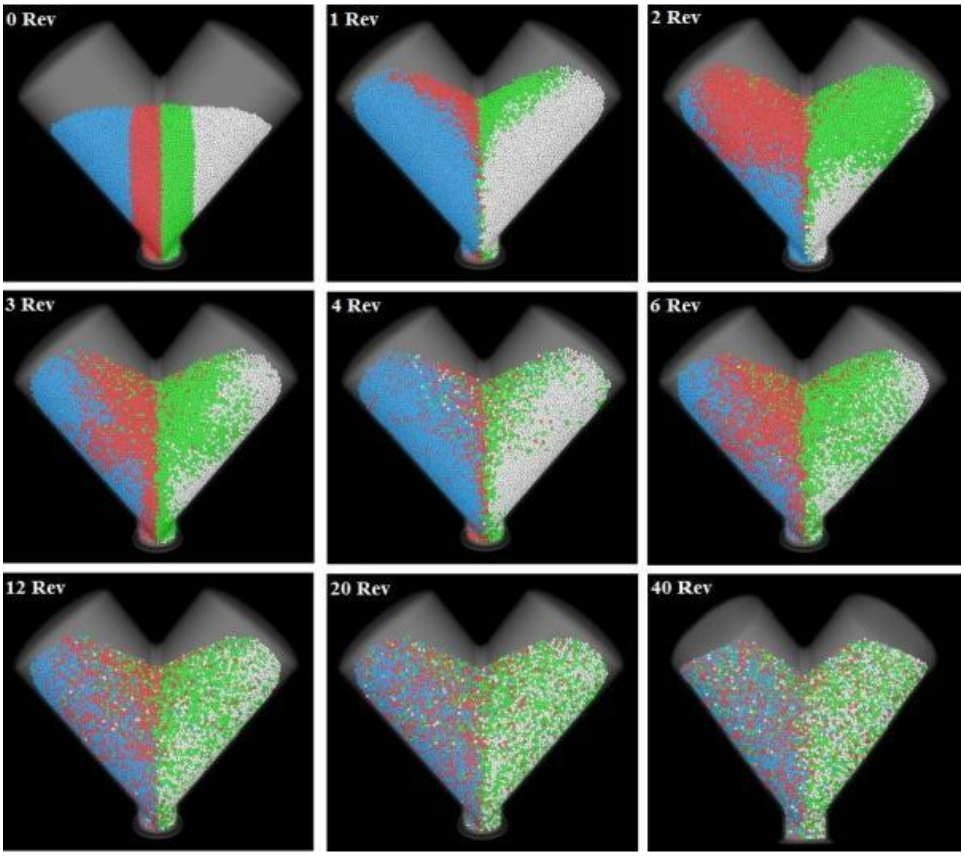

Figure 14: Mixing within a V-blender. The particle bed (consisting of identical particles) was divided into four sections according to volume and each section was given a different colour. It can be seen that very few mixer revolutions result in good mixing of the two colours contained within each leg, but poor mixing between the legs.

Cleary & Sinnott (2008) suggested that the particle migration rate across the legs of the V-blender could be improved if the symmetry of the V was changed and therefore the trend of circular motion was broken.

Plough share mixer:

A plough share mixer is a device consisting of an outer shell or cylindrical drum. Instead of the drum rotating as in a drum mixer, a shaft is placed through the length of the drum and fitted with one or multiple plough shaped blades. These blades are set up to have minimal clearance between the blades and the container wall. As the shaft rotates, each blade will pass through the material in the container creating a void behind the blade. As material flow into the void, diffusion of the material takes place, causing mixing.

As the blade is driven through the material, it will also push material in the direction of rotation. This, together with the flow into the void behind the blade, creates circulation patterns on either side of the plough, causing additional mixing.

Cleary & Sinnott (2008) used DEM to simulate a plough share mixer and their results correspond with the work conducted by Laurent *et al.* (2004), who used PEPT to track the movements of particles within a plough share mixer.

Cleary & Sinnott (2008) found that, if one blade is used, the device is capable of relatively fast radial mixing times, but that the radial mixing is hindered by slow axial mixing. The axial mixing does however increase with an increase in blade rotation rate. When multiple blades are used, the axial mixing improves, because the circular flow patterns of the material caused by the blades can interfere with each other causing better particle dispersion. There are unfortunately still stagnant zones where less mixing occurs due to the fact that the number of blades is limited.

Figure 15 and Figure 16 both show radial and axial particle dispersion within a single- and multiple-bladed mixer respectively. From Figure 15 and Figure 16, it is clear that an increase in the number of blades/ploughs, will give rise to an increase in the rate of axial dispersion and mixing.

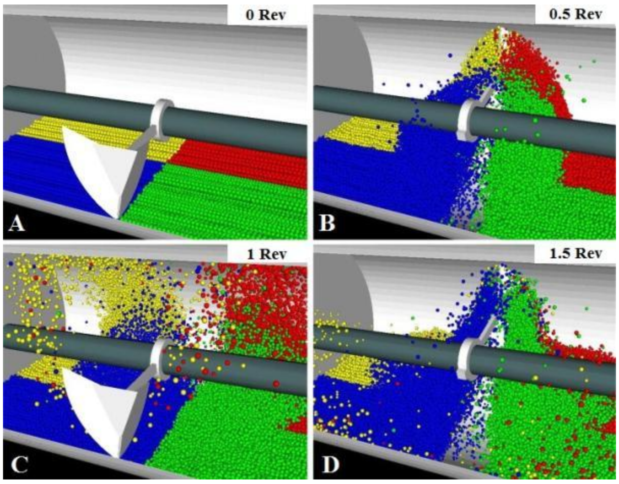

Figure 15: The axial and radial dispersion within a single bladed mixer after 0, 0.5, 1 and 1.5 revolutions respectively.

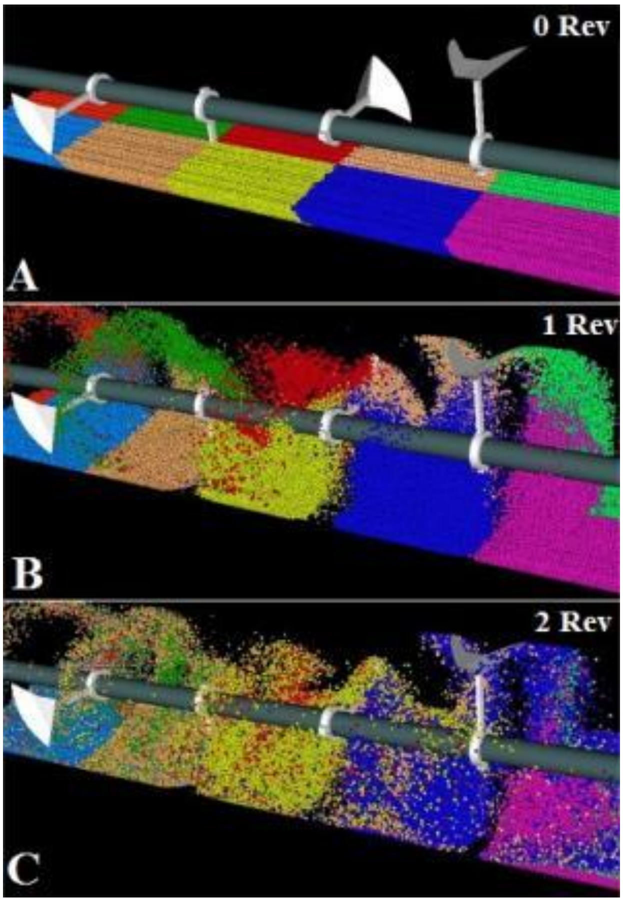

Figure 16: The axial and radial dispersion within a multiple bladed mixer after 0, 1 and 2 revolutions respectively.

Peg mixer:

This device is also a continuous mixer and follows a similar approach as a drum mixer, except in this case the drum/cylinder is kept stationary and a rotating shaft, fitted with pegs/pins, provides the material agitation. A number of pins are arranged in a spiral pattern along the length of the rotating shaft.

Radial mixing is obtained by the pegs similar to that created by the blades in a plough share mixer. The difference between the two is that, in the case of the peg mixer, the material flow and mixing can be controlled in the axial direction due to the spiral arrangement of the pins - assuming that the viscosity of the material is high enough. As material is fed into the device it is gradually conveyed through the device whilst being mixed both in the radial and axial direction.

Cleary & Sinnott (2008) found that this device had reasonable mixing characteristics, but the peg mixer tends to promote size segregation within the mixture. As material is pushed forward by the pegs, the smaller particles manage to pass through the voids and only the larger particles are driven forward, causing the larger particle to flow in the direction of rotation and smaller particles to flow in the opposite direction.

Figure 17 shows how the pegs are capable of driving the material axially through the mixer and Figure 18 shows how, as the shaft rotates clockwise, the larger particles are found to settle in the direction of rotation and the smaller particles settle against the direction of rotation. This clearly illustrates size segregation.

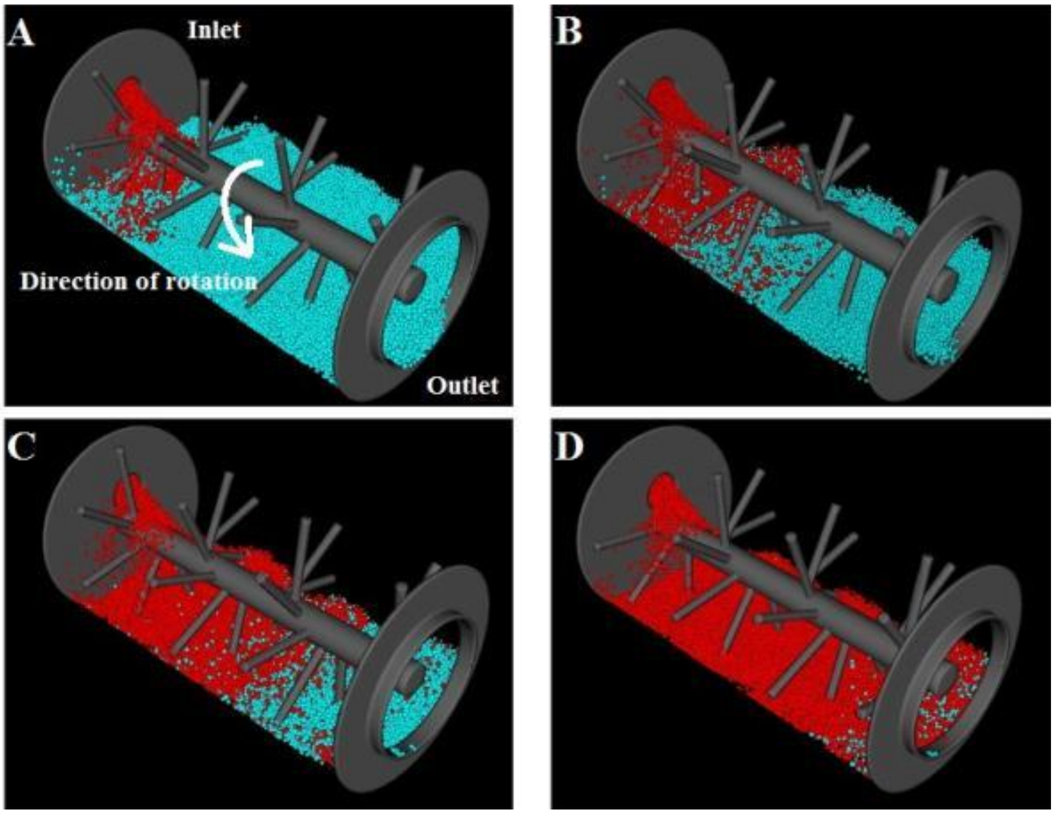

Figure 17: A demonstration of the driving mechanism created by the pegs within a peg mixer. In Part A material is once again fed into the mixer (Similar to Figure 12.A). As the material is driven through the mixer in Part B and part C, it can be seen that the line separating the two sets of material is distorted to a larger extent than the case of the drum mixer.

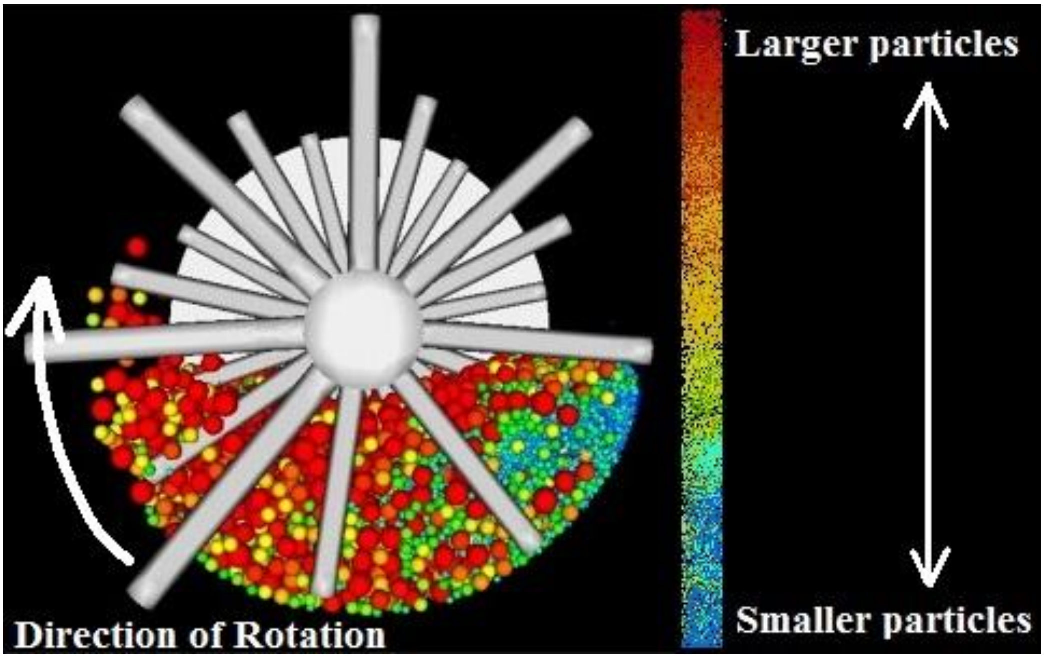

Figure 18: A representation of the segregation that can occur within a peg mixer. Large particles (Red) can be expected to flow in the direction of rotation, whereas the smaller particles (blue) can be expected to flow against the direction of rotation.

Disc- and blade-impeller mixers:

Both of these devices consist of a stationary vertical cylinder. For the disc impeller, a rotating disc forms the bottom of the cylinder. The cylinder is filled with granular material and, as the disc rotates, the friction between the material and the container wall is used as the primary mechanism to induce mixing.

For the blade impeller, a rotating blade is fitted to the bottom of the cylinder, where the friction between the base and the material as well as the friction between the container wall and the material is used as the mechanisms to induce mixing.

According to Cleary & Sinnott (2008) the motion of the disc impeller initially creates material flow patterns, but is unable to induce radial mixing. Initially the friction between the particles and the container wall is adequate to restrain the particles nearest to the container wall inducing some mixing, but the centre mass of particles will rotate on the disc as a solid body.

The blade impeller is able to agitate the particles at the container wall as well as at the centre, therefore overcoming the solid body problem. The blade impeller is therefore capable of both radial mixing and mixing azimuthally.

Figure 19 shows a segment of material within a disc impeller mixer. It is shown here how the friction between the particles and the mixer wall initially causes some mixing, but eventually leads towards the formation of concentric particle rings where very little radial mixing takes place. Figure 20 shows a segment of material being mixed with a blade impeller mixer. It shows that the material is mixed both in the radial direction and azimuthally.

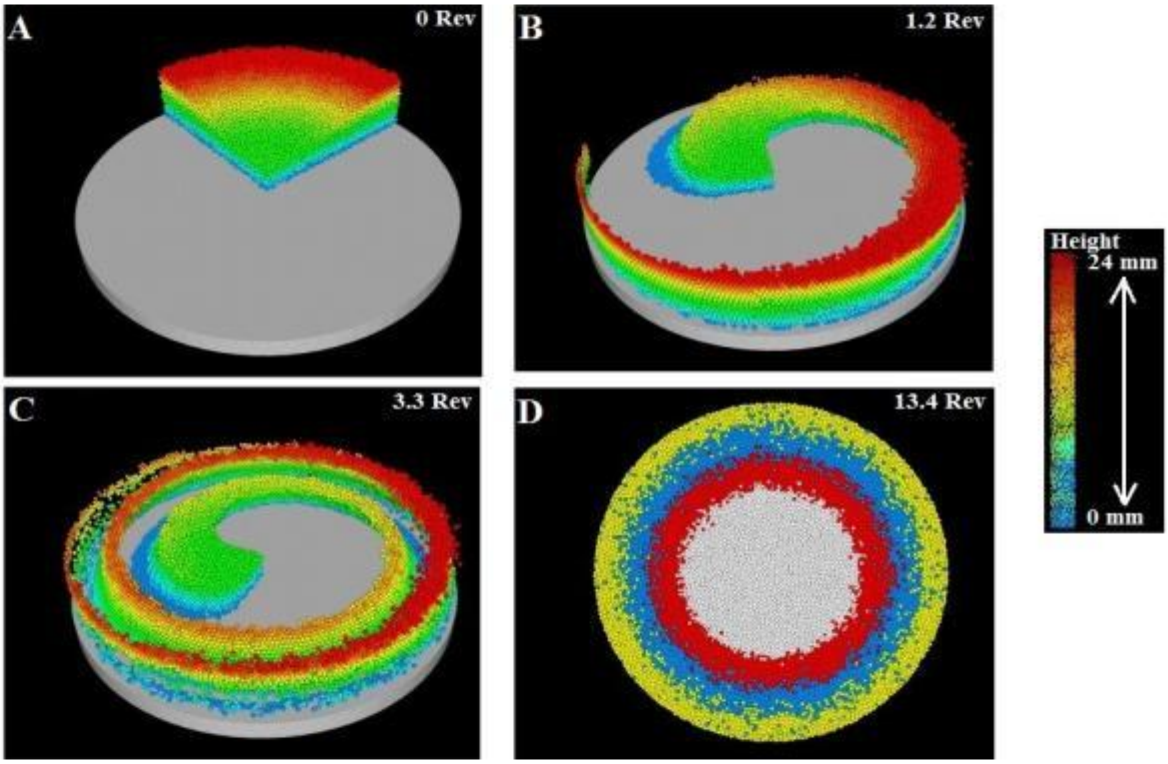

Figure 19: Particle flow patterns within a disc impeller mixer. One quadrant of material in the mixer is shown in Parts A, B and C, where the particles are coloured according to their height within the quadrant. The particles within the entire mixer is shown in Part D, where the concentric circles are visible as described in the text,

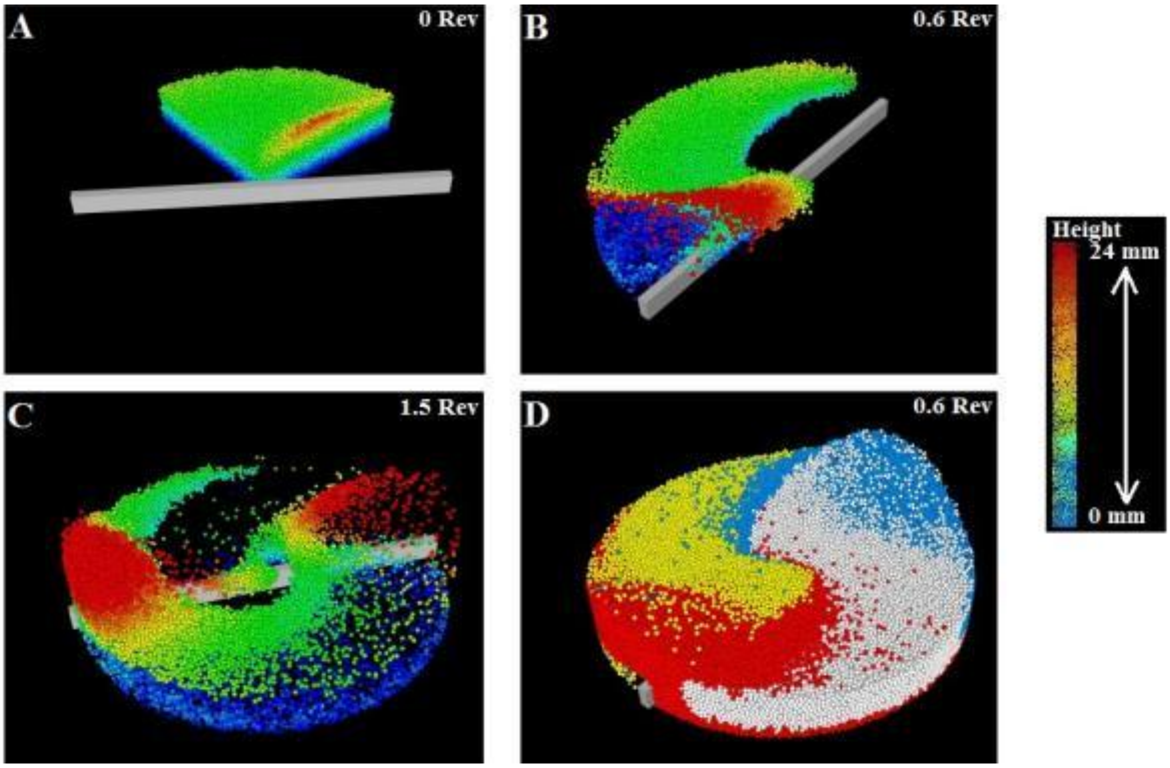

Figure 20: Particle flow patterns within a bladed impeller mixer. One quadrant of material in the mixer is shown in Parts A, B and C, where the particles are coloured according to their height within the quadrant. The particles within the entire mixer are shown in Part D. Here it can be seen that mixing takes place in the azimuthal and radial direction.

2.6. Conclusion and Scope

The literature study showed that variations occur in mixing effectiveness and that this effectiveness is strongly influenced by specific properties of the mixing system. Table 2 is a summarized representation of these variations and their effects. It shows the variable properties in the left-hand column, the possible variations in the middle column and a summary of the effects in the right-hand column.

Table 2: Summary of the effect of variations in granular material mixing systems.

Property	Variations *	Effect of variations
Container shape	Circular Non-Circular	The level of chaotic advection and therefore mixing is increased in tumbling mixers as the geometry of the mixer is made less circular.
Degree of filling	Less than half Exactly half More than half	The mixing per rotation is increased as the fill level is increased up to a certain point. Any further increase in fill level will hinder the particle flow and therefore result in a decrease in mixing per rotation. Thus producing an optimum fill level, varying for different mixers.
Mechanism of mixing	Avalanching Rolling/Cascading Cataracting Centrifuging	The mechanism by which the particles interact within the mixer greatly influences the degree to which the particles are dispersed and is managed by the speed of rotation. The best mixing per rotation occurs with a cataracting flow mechanism, whereas the centrifuging flow mechanism produces the least mixing per rotation. An optimum speed of rotation can therefore be found for each of the various mixers.
Properties of particles	Identical powders Density variations Size variations	Size and density variations in particles lead to material segregation. It is difficult to simulate both size and density variations at the same time and therefore the materials to be mixed should be selected to either have the same particle size or weight to ease mixing calculations/simulations. When material with extremely diverse particle properties are to be mixed, a buffer material can be added to the recipe to minimize this problem. Segregation during mixing is acceptable to a certain extent, as it can assist during some mixing processes, but it should be kept to a minimum as it can also lead to a blend becoming unmixed during handling after the mixing process.

* Possible variations that can occur or are available

During the literature study two types of mixers were identified namely continuous and batch mixers. It is clear from the literature study that the common problems of continuous mixers include plug flow formation and material segregation. This gives rise to an uneven blend homogeneity which, from the problem statement, is unacceptable.

The literature study also identified the three mixer classes as gravity driven, stirred and high shear mixers. It was shown that the evaluated stirred mixers achieved a good level of mixing in the tangential and radial direction, but less in the axial direction. This influences the blend homogeneity and is therefore unwanted. A similar occurrence was also found in the high shear disc-impeller mixer that was evaluated.

This review also indicated that a gravity driven mixer typically had a lower capital investment as compared to a stirred mixer of the same size. The optimum fill level of a gravity driven mixer was also generally higher than that of a stirred or high shear mixer, but the gravity driven mixer requires more revolutions to achieve the same blend homogeneity as a high shear mixer.

It can therefore be said that, for a certain throughput, a gravity driven mixer will typically produce a small quantity of large, well-mixed batches, whereas a high shear mixer would produce a larger quantity of smaller, well mixed batches.

Cleary & Sinnott (2008) suggested a modification to the symmetry of the legs of the V-blender as shown in Figure 21. Figure 21 shows a diagram of the standard V-blender in Part A and a diagram of the modified V-blender in Part B.

They suggested that, by changing the symmetry of the legs of the mixer, an uneven amount of material will flow into each of the legs of the mixer as it rotates. Through this alteration, it is suggested that the rate of mixing between the mixer legs will increase and that the mixing system will now be capable of producing a larger quantity of large batches in a smaller time period. *The size of the mixing system can therefore be reduced for the same throughput, making a modified V-blender a mixing system that can possibly meet the size restriction set in the aim of the project.*

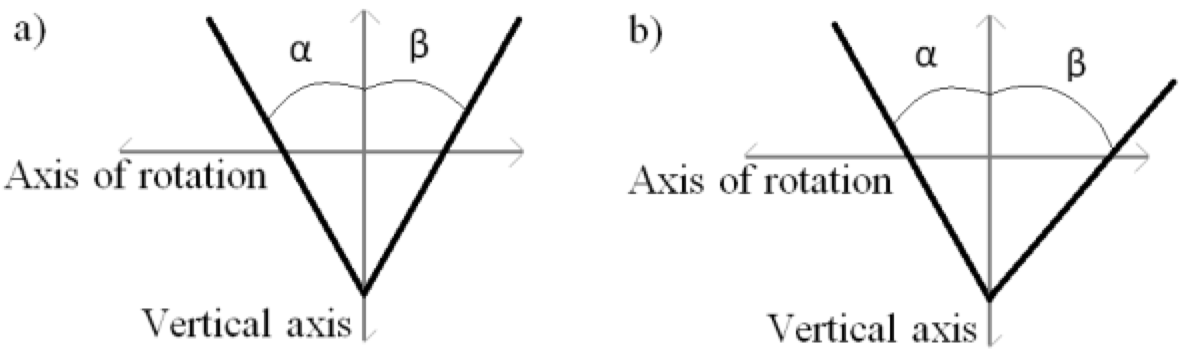

Figure 21: The schematic diagram for a V-blender is given in Part A, with α=β. The modified V-blender as suggested by the literature is given in Part B, with α≠β.

The problem with the suggested modification to the V-blender is that the dissymmetry of the mixer will make it look unattractive, but more importantly, it will be unbalanced under load.

Having considered the findings of the review, it is thought that by adding a third leg to the V-blender – thereby producing a Y shape – the same result as changing the symmetry of the two legs, as suggested by Cleary & Sinnott (2008), can be achieved. This modification could then lead to improved mixing effectiveness at reduced lead times with less complicated mechanics.

A schematic diagram of the suggested Y-shaped mixer, now dubbed a Y-blender, is presented in Figure 22.

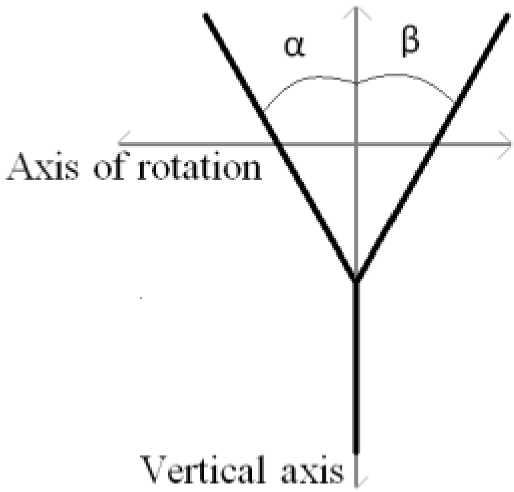

Figure 22: The schematic diagram for the derived Y-blender with α=β.

Because the Y-blender is introduced as a new concept through this study, there is no data to confirm the effectiveness of such a mixer. For this reason this study also evaluated the performance of the suggested Y-blender compared to a standard V-blender in an effort to select and design a mixer that would conform to the aim of the study and the user specification requirements for a compact transportable mixing system.

In addition, the study produced a control algorithm in order to evaluate the feasibility of programmable control hardware applied to an assembly of the designed mixing system, which was produced with the aid of rapid prototyped models.

Chapter 3: Evaluation of the mixing effectiveness

In this chapter the effectiveness of the suggested Y-blender is compared to that of a typical V-blender. Three methods are presented to determine the effectiveness of mixing within the mixers. The experimental procedures followed by each of the presented methods are described individually, along with the results of the respective procedure.

3.1. Mixer setup

As concluded through the literature survey, a modified V-blender – referred to as a Y-blender – could be used to achieve better mixing results. Such a blender, if found effective, would ideally suit the design requirements for the transportable mixing and blending system.

3D CAD models of a V-blender and the suggested Y-blender that was used for evaluation purposes are presented in Figure 23 and Figure 24. The blenders (mixers) were produced from 3.2mm thick medium density fibreboard (MDF), otherwise known as SupaWood, by laser cutting. Detailed drawings of each of the mixers are provided in Appendix C – Detail drawings of the V- and Y-blenders and supporting frame.

Both mixers were designed to have a volume of 7635cm^3 +/- 5%. The angles between the symmetrical legs of the mixers are 60^o. The mixers were also fitted with sight glasses made from 3mm clear Perspex in order to observe the fill level within the mixers, as well as the mixing mechanism within the mixer.

A single frame was manufactured to support either the V- or the Y-blender. The frame was fitted with a direct current (dc), worm gear motor with an output shaft speed of 50 revolutions per minute (rpm) at 12V. A computer power supply connected to the dc motor ensured that the rotational speed of the motor shaft was kept constant.

As the manufactured V- and Y-blenders were fitted to the frame, a laser cut gear system – made from 6mm Perspex and a gear ratio of 8:25 – was used to connect the respective mixer shaft to the output shaft of the dc motor. This gear ratio of 8:25, along with the gears within the motor enclosure, allowed the mixers to achieve a rotational speed of 16rpm.

Each mixer was fitted with a sliding door with rubber seals to allow for material to be added to, or removed from, the mixer with ease.

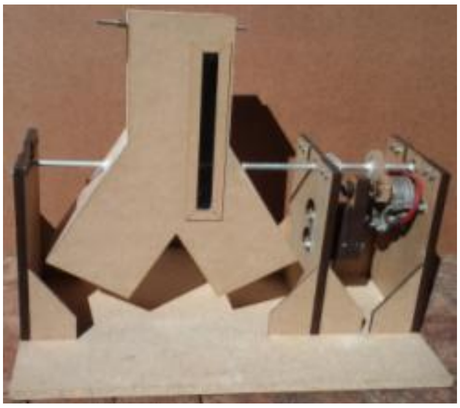

Figure 23: The model Y-blender, shown in the filling position, as used during the experiments.

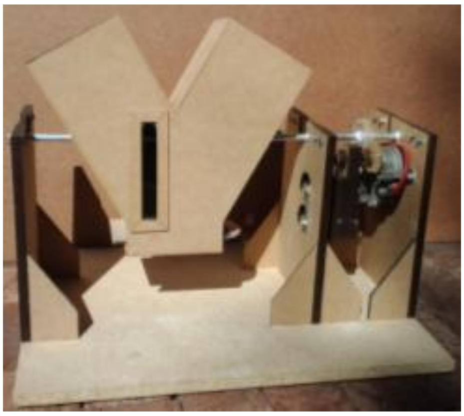

Figure 24: The model V-blender, shown in the unloading position, as used during the experiments.

3.2. Experimental procedure

To evaluate the effectiveness of the two mixers, the level to which the mixers were filled was varied.

A simple recipe, consisting of only three ingredients was used for the experiments. The recipe consisted of 87% fine yellow maize flour, 12.75% sugar and 0.25% colorant.

The three ingredients used, simulated a wide variety of the particle properties that were shown by the literature study to affect mixing. These properties included variations in particle shape, size and density. These variations relates closely to those found in the recipes in the industry and could therefore be used to compare the mixing characteristics of the two mixers.

At first, both mixers were filled to 30% of their capacity with ingredients according to the above mentioned recipe. Each of the mixers was then rotated for 100 revolutions and samples were taken after 2, 4, 6, 8, 10, 15, 20, 25, 30, 35, 40, 50, 60, 80 and 100 revolutions respectively. After this, the mixers were filled to 60% of their capacity with the same ratio of ingredients. The same mixing and sampling procedure was then carried out at this increased fill level.

Various procedures, which are described in the following sections, were then carried out to study the influence of the increase in fill level on the level of mixing within the acquired samples.

Figure 25 and Figure 26 show models of the V- and Y-blenders respectively, along with the axis of rotation of the mixers as used during the above mentioned procedures.

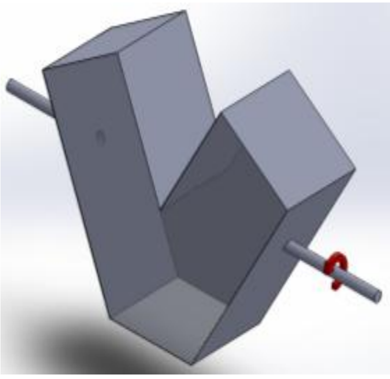

Figure 25: The concept of a V-blender showing the axis of rotation.

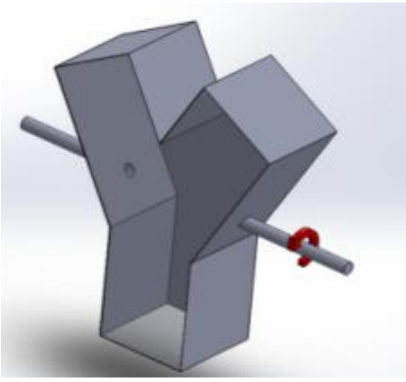

Figure 26: The concept of a Y-blender showing X as the axis of rotation.

3.3. Discussion of evaluation procedures

3.3.1. Evaluation of the mixer effectiveness using a Scanning Electron Microscope

The first concept to evaluate the level of mixing within the samples that were taken from the mixture as described in the previous section was to use a Scanning Electron Microscope to photograph the samples. These photographs would then be used to count the respective number of maize, sugar and colorant particles within the sample. After examining several samples, the number of rotations required to achieve an acceptable level of mixing could then be identified.

In Figure 27, a sugar particle with a diameter of roughly 1mm is highlighted, and in Figure 28, two colorant particles with diameters of roughly 50µm are also highlighted. These figures are photographs taken by a Scanning Electron Microscope of some of the samples taken from the mixtures.

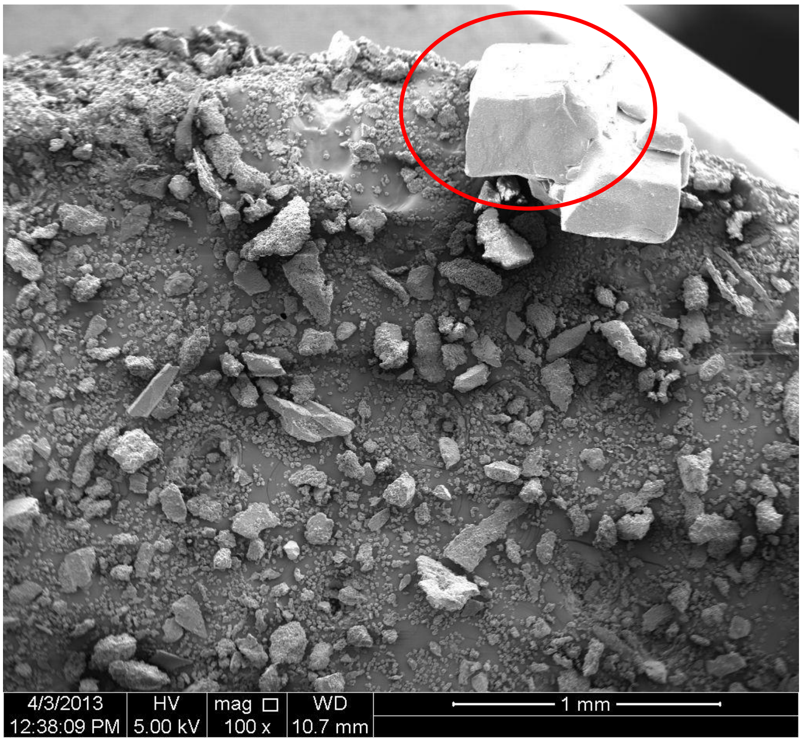

Figure 27: Magnification of particles within a sample of the mixture, highlighting a sugar particle roughly 1mm in diameter.

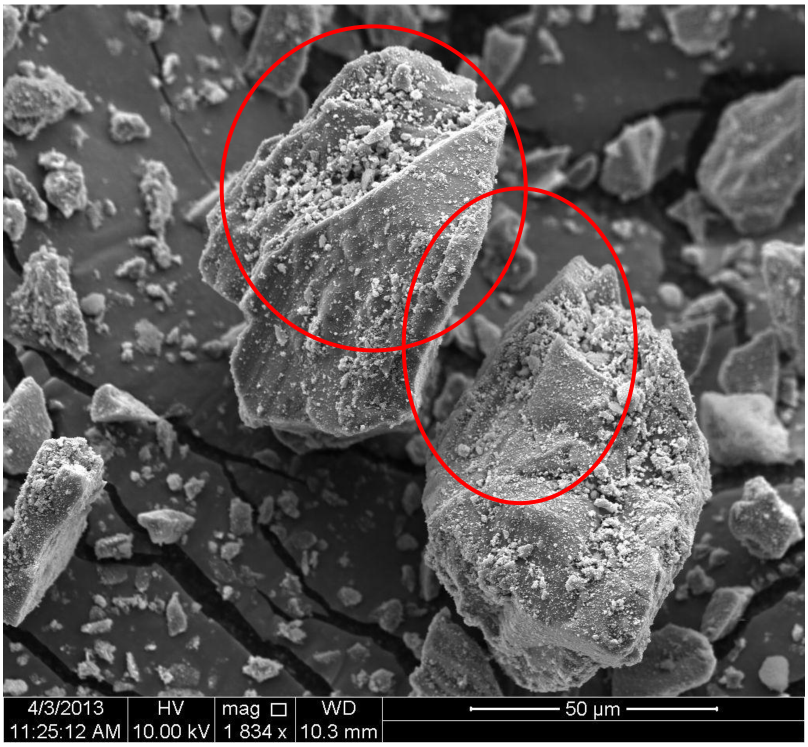

Figure 28: Magnification of particles within a sample of the mixture, highlighting two colorant particles roughly 50µm in diameter.

In Figure 29 a photograph of a maize sample is given as taken by a Scanning Electron Microscope not different to that of Figure 27 and Figure 28. The large particle in the centre of this photograph is a rather large piece of a maize kernel, whereas the smaller particles around it are pieces of the starch that these kernels consist of.

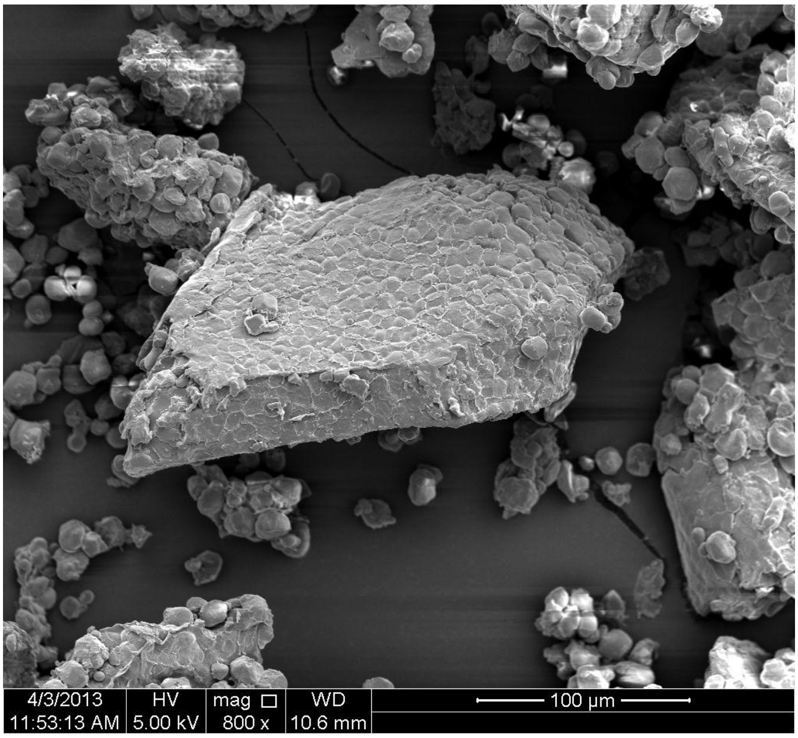

Figure 29: Magnified sample of maize flour showing variations in particle shape and size.

After studying Figure 27 and Figure 28, it was found that the difference in the average particle size of the sugar and the colorant was so great that a single reliable sample could not be taken containing both substances.

The problem discovered through Figure 29 led to the conclusion that the particle shape and size of each individual substance also varied so widely that it was not feasible to quantify the amount of particles from each substance.

Due to the uncertainty caused by these two problems, it was therefore concluded that it was not feasible to compare the mixing effectiveness of the two mixers using a Scanning Electron Microscope.

3.3.2. Evaluation of the mixer effectiveness using a filtration procedure

It is known that maize is only soluble in water at extremely high temperatures and pressures. It was found that the colorant used during these experiments, as well as sugar, could dissolve in water well below the boiling point of water.

A filtration method could therefore be considered to determine the ratio of maize vs. the combination of sugar and colorant within each of the samples obtained during the experimental procedure. This ratio could in turn be used as an indication of the level of mixing or blend uniformity within the samples

An experiment was accordingly set up where samples were produced as described within the section covering the experimental setup. At each of the described sampling intervals, four samples were taken. The first sample taken was tabulated within sample set one, with the second samples drawn being tabulated within a second sample set. The same was also done to produce third and fourth sample sets.

Careful consideration was required to determine the size of the samples. If a sample size was too small, for instance if a single particle weighed 0.01g and the sample size was 0.01g, the results would be biased. The opposite is also true where the sample was too large, for example where the entire batch was 500g and the sample was 400g, the result would also be influenced. A reasonable sample size was therefore required, where the sample should be as large as was practical, without causing the results to be bias.

For the recipe under consideration, a sample of 3.26g was less than 0.5% of the total mixture, indicating that it was small enough not to cause the results to become biased.

A Sample of 3.26g of the recipe amounted to 25ml in volume. As presented above, the particle sizes of the ingredients varied between 50µm for the colorant and 1mm for the sugar. A 3.26g sample therefore was large enough to accommodate multiple particles of each ingredient.

An average sample size of 3.26g was therefore used throughout the experiment and considered appropriate.

The acquired samples were then individually stored within marked filters that were weighed and tabulated after being dried through the use of high energy light sources.

The sample, containing the sugar, colorant and maize, was dissolved in water at room temperature and the maize was filtered out, separating it from the sugar and colorant. The filters, containing only maize, were then once again dried, weighed and tabulated.

The tabulated data produced by this procedure is presented in Appendix D – Data acquired during filtration evaluation method.

To compare the mixing effectiveness of the V- and Y-blender, the definition of the effectiveness of mixing presented in Section 2.2 of the literature survey was used.

The equation

$$s = \sqrt{\sum \frac{(x_i - \bar{x})}{n-1}},$$

was used to calculate the standard deviation between the four samples taken at each interval, as suggested by Smith (2011) in Section 2.2 of the literature survey. The equation

$$s_0 = \sqrt{p(1-p)},$$

was used to calculate the standard deviation of all the material within the mixer as also suggested by Smith (2011).

This value was updated after every rotation, because the ratio of material left within the mixer changed as samples were extracted from the mixer. To calculate the mixing index (M) at every interval, equation (2),

$$M = \frac{s - s_\infty}{s_0 - s_\infty},$$

of Section 2.2 of the literature survey was used with an acceptable deviation limit of $s_\infty = 0.002$.

By using these equations, the results shown in Figure 30 to Figure 33 were generated. The data used to generate Figure 30 to Figure 33 can also be found in Appendix D – Data acquired during filtration evaluation method.

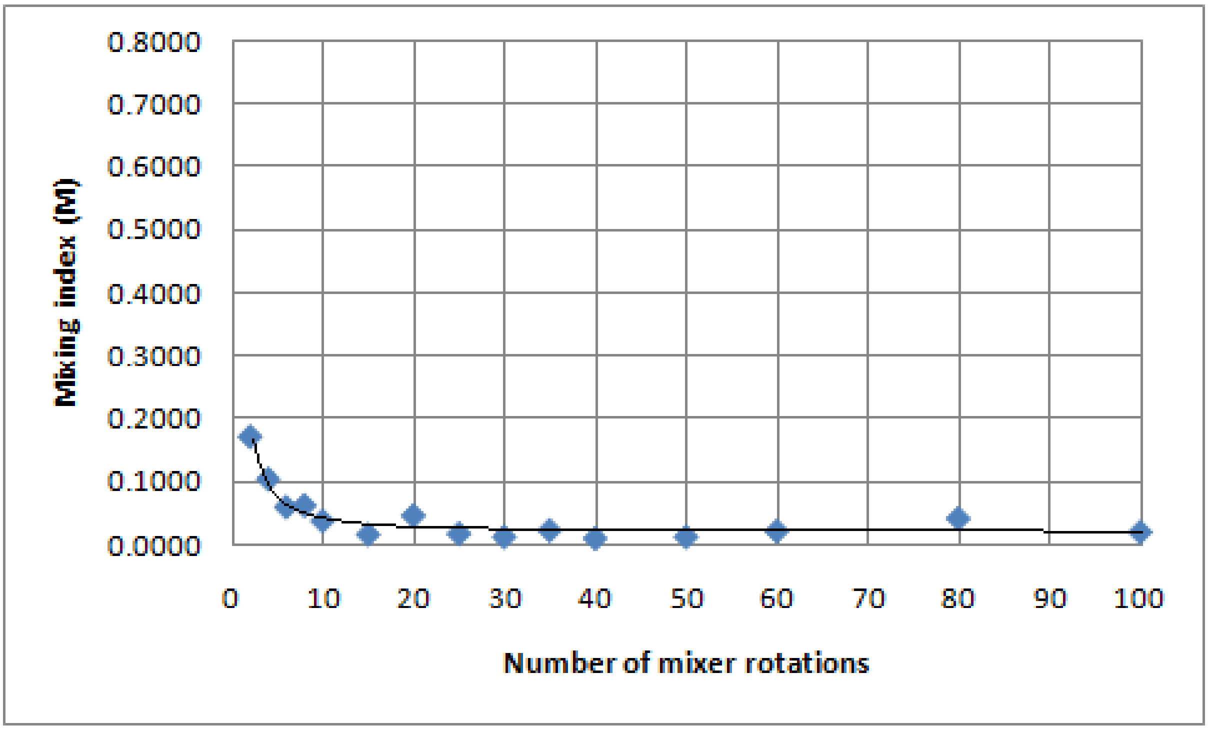

Figure 30: Mixing index per revolution for the Y-blender with a fill level of 30% (Discussion of figure provided in text).

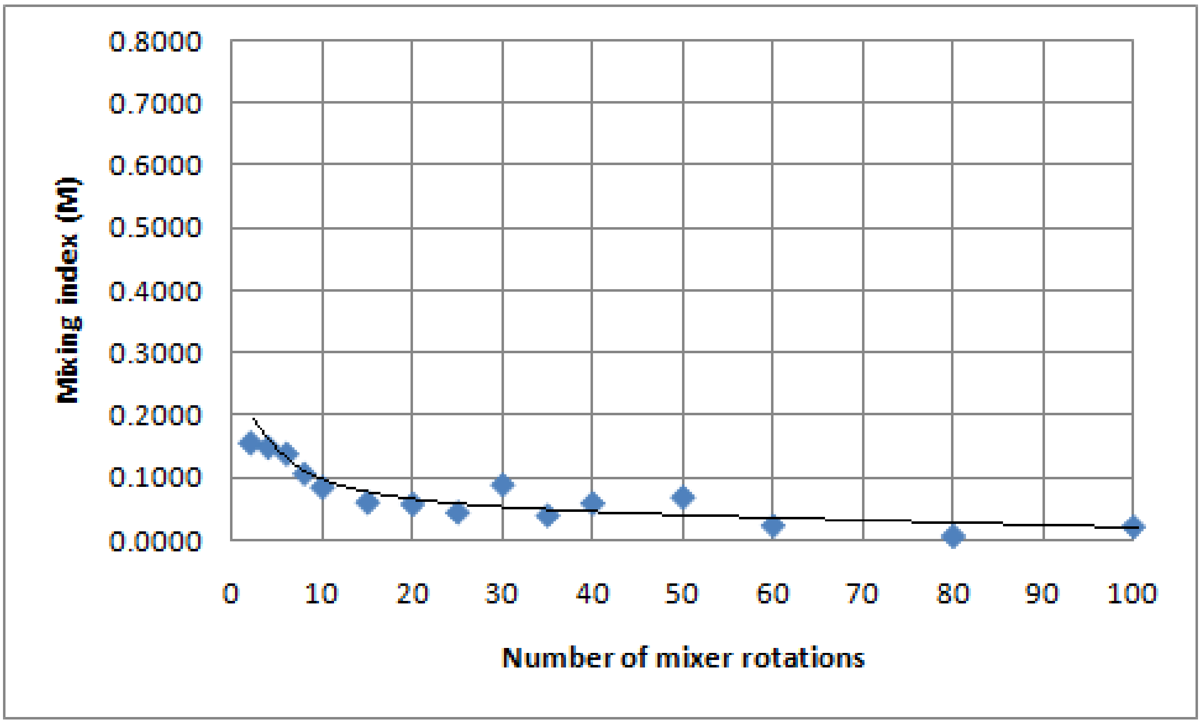

Figure 31: Mixing index per revolution for the V-blender with a fill level of 30% (Discussion of figure provided in text).

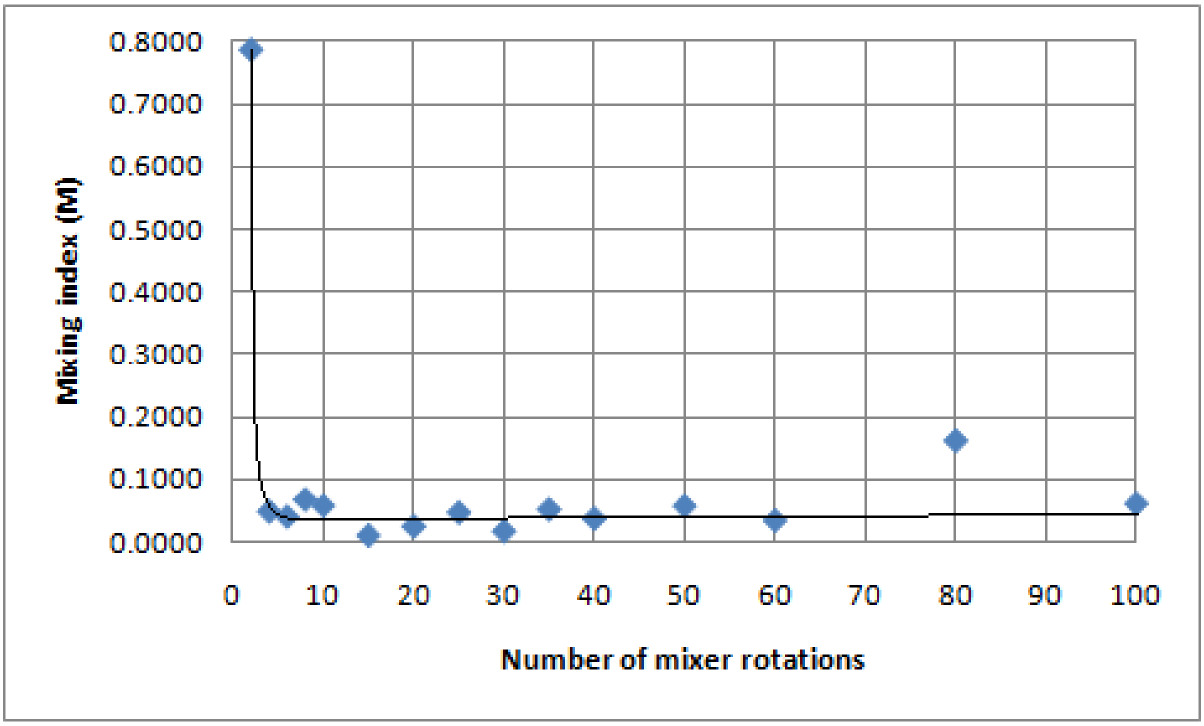

Figure 32: Mixing index per revolution for the Y-blender with a fill level of 60% (Discussion of figure provided in text).

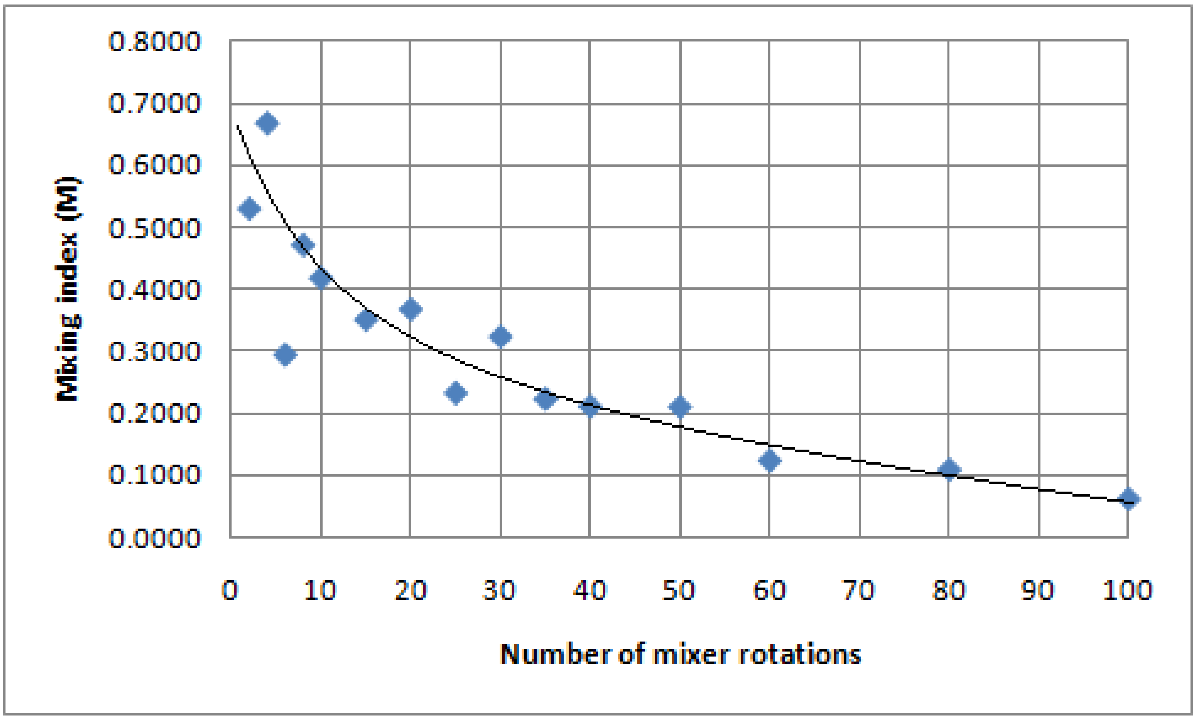

Figure 33: Mixing index per revolution for the V-blender with a fill level of 60% (Discussion of figure provided in text).

For the purposes of this study, a Mixing index (M) of less than 0.1 was selected as overall blend uniformity. After evaluating the results presented in Figure 30 to Figure 33, it can be concluded that the Y-blender achieved this level of mixing within the first ten mixer revolutions for both fill levels. The V-blender is also capable of achieving this level of mixing within the same time period for a fill level of 30%, but the same cannot be said for the V-blender with a fill level of 60%. (By investigating

Figure 33, it is found that a Mixing index of less than 0.1 is only accomplished after a hundred revolutions).

Through these findings, it can therefore be concluded that the evaluated Y-blender is more versatile than the evaluated V-blender, as it can maintain the same efficiency, to a certain extent, irrespective of fill level. This implies that a smaller Y-blender can deliver the same throughput as a larger V-blender.

3.3.3. Evaluation of the mixer effectiveness using a camera and computer software

To validate the results and conclusions of the filtration evaluation procedure, a similar approach was followed to that suggested by Aissa *et al.* (2011) in Section 2.2 of the literature survey.

As described in the experimental setup section, a mixture consisting of 87% fine yellow maize flour, 12.75% sugar and 0.25% colorant was used during this procedure.

A 3.15 Megapixel camera was set up in a fixed position and used to take a 2048 X 1536 pixel photograph of the mixture within the mixer at the sampling intervals stated within the experimental setup section. Because the sampling could therefore take place without disturbing the material within the mixer, this method of sampling can be seen as a nonintrusive method of sampling.

Two random areas of 150 pixels wide by 150 pixels high were then selected on each photograph and used during the rest of this evaluation procedure. These selected areas are therefore – for this procedure – referred to as the two samples taken at each sampling interval.

Each sample was then converted to grey-scale, after which a histogram of the colour distribution was obtained for each spectrum. This was then used to calculate the standard deviation of the mean intensity of the pixels within each sample.

This standard deviation provides an indication of the degree of mixing that took place in the same manner as the ‘Area fraction’, as applied by Aissa *et al.* (2011) in Section 2.2 of the literature survey.

When reviewing the standard deviation within each of the samples, one must remember that different materials of different colours are used. If it was a liquid being mixed, the colours would be blended, resulting in a standard deviation equal to zero. As these are solid materials, the colours would not reach point uniformity, as described in Section 2.2 of the literature survey, but only overall uniformity. This will result in a standard deviation, within each sample, of greater than zero.

To overcome this, the standard deviations across the two samples taken at every interval were compared. The difference between the standard deviations across the two samples would therefore become smaller as blend uniformity was reached.

The trends of the standard deviations produced by these experiments could therefore be compared to the trends found during the filtration procedure.

These differences were tabulated for each of the intervals observed in Figure 34 to Figure 37. The data used to generate Figure 34 to Figure 37 can be found in

Table 3 to Table 6.

Table 3: Data produced by the image analysis evaluation method for the Y-blender filled to 30%.

Y-Blender @ 30% fill			
Standard deviation within sample set 1	Standard deviation within sample set 2	Difference in standard deviation	Number of revolutions
4.73	3.98	0.75	2
4.90	3.34	1.56	4
3.02	3.57	0.55	6
3.40	3.26	0.14	8
3.02	2.63	0.39	10
3.37	2.57	0.80	15
4.06	3.42	0.64	20
2.82	3.62	0.80	25
3.06	2.74	0.32	30
2.75	3.23	0.48	35
3.46	3.80	0.34	40
3.36	4.17	0.81	50
2.61	2.95	0.34	60
3.68	3.35	0.33	80
2.95	3.20	0.25	100

Table 4: Data produced by the image analysis evaluation method for the Y-blender filled to 60%.

Y-Blender @ 60% fill			
Standard deviation within sample set 1	Standard deviation within sample set 2	Difference in standard deviation	Number of revolutions
3.13	11.72	8.59	2
4.68	4.42	0.26	4
2.53	3.78	1.25	6
2.83	3.62	0.79	8
2.34	2.92	0.58	10
2.86	2.74	0.12	15
2.97	3.44	0.47	20
2.80	3.56	0.76	25
3.38	3.30	0.08	30
3.45	3.03	0.42	35
3.62	3.20	0.42	40
3.32	3.41	0.09	50
3.30	3.26	0.04	60
3.06	2.77	0.29	80
3.04	3.05	0.01	100

Table 5: Data produced by the image analysis evaluation method for the V-blender filled to 30%.

V-Blender @ 30% fill			
Standard deviation within sample set 1	Standard deviation within sample set 2	Difference in standard deviation	Number of revolutions
4.69	8.22	3.53	2
4.27	5.80	1.53	4
4.68	3.93	0.75	6
3.51	3.59	0.08	8
3.84	3.30	0.54	10
4.34	4.47	0.13	15
3.19	3.96	0.77	20
3.42	4.03	0.61	25
3.08	3.73	0.65	30
3.47	4.10	0.63	35
3.15	3.69	0.54	40
3.78	3.64	0.14	50
4.54	4.08	0.46	60
4.31	4.59	0.28	80
4.31	3.27	1.04	100

Table 6: Data produced by the image analysis evaluation method for the V-blender filled to 60%.

V-Blender @ 60% fill			
Standard deviation within sample set 1	Standard deviation within sample set 2	Difference in standard deviation	Number of revolutions
6.19	5.28	0.91	2
4.14	5.43	1.29	4
4.25	5.69	1.44	6
3.51	6.38	2.87	8
3.03	4.71	1.68	10
3.88	4.97	1.09	15
3.27	4.79	1.52	20
3.21	3.80	0.59	25
3.89	3.38	0.51	30
3.37	4.21	0.84	35
2.79	3.98	1.19	40
3.51	4.03	0.52	50
4.06	4.10	0.04	60
3.06	4.35	1.29	80
3.63	3.58	0.05	100

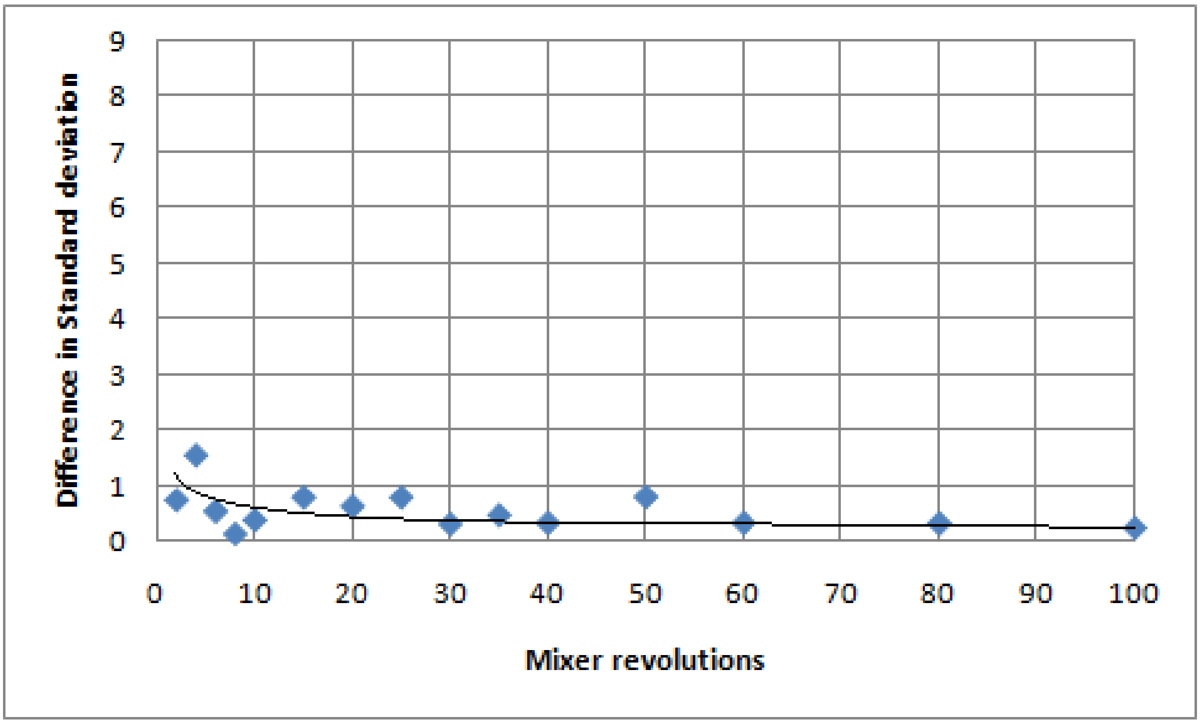

Figure 34: Difference in standard deviation between the two samples taken at every interval for the Y-blender filled to 30% (Discussion of figure provided in text).

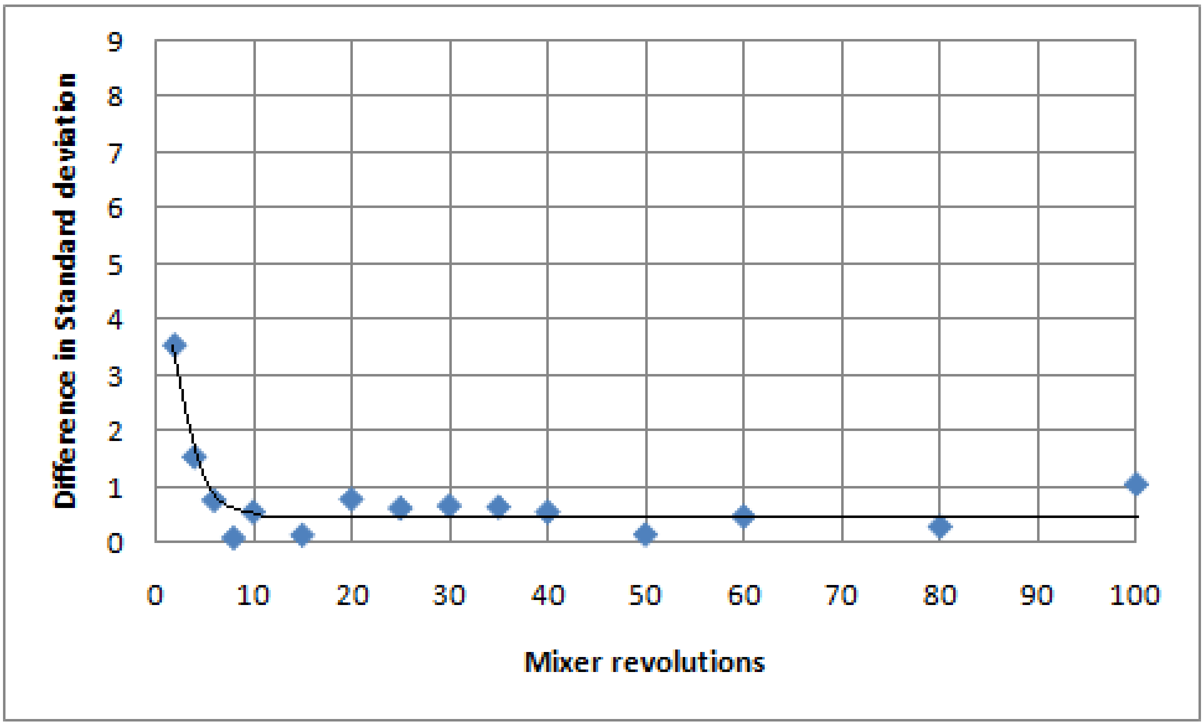

Figure 35: Difference in standard deviation between the two samples taken at every interval for the V-blender filled to 30% (Discussion of figure provided in text).

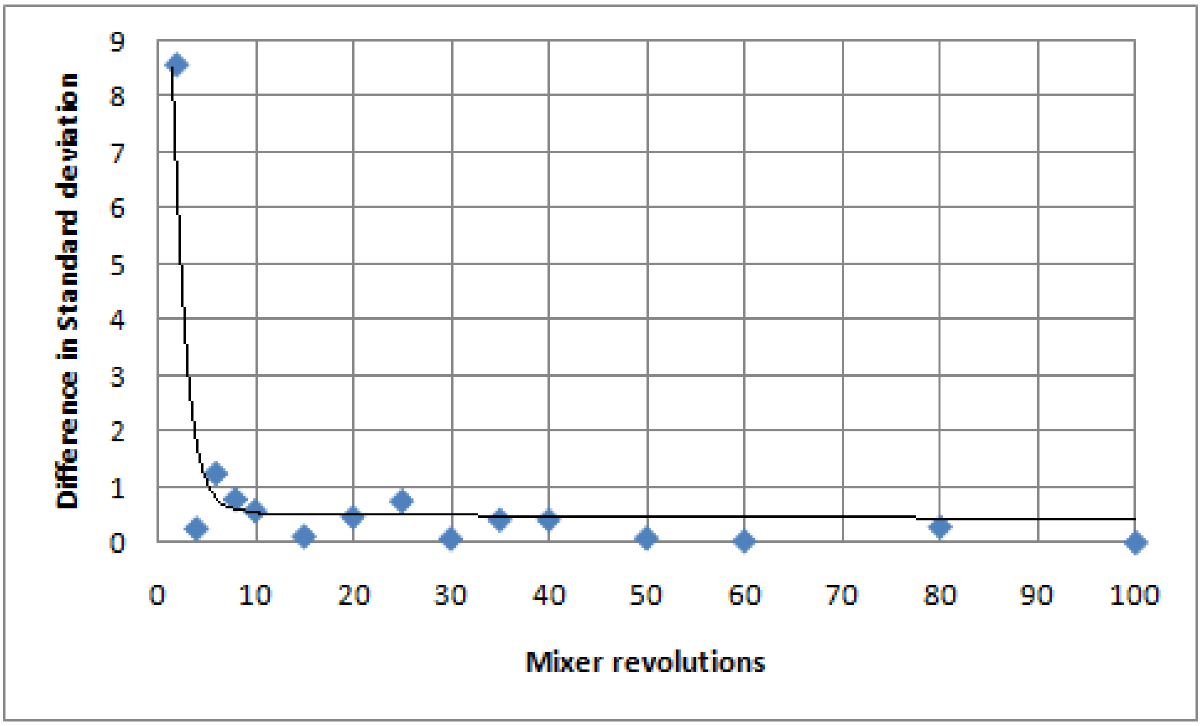

Figure 36: Difference in standard deviation between the two samples taken at every interval for the Y-blender filled to 60% (Discussion of figure provided in text).

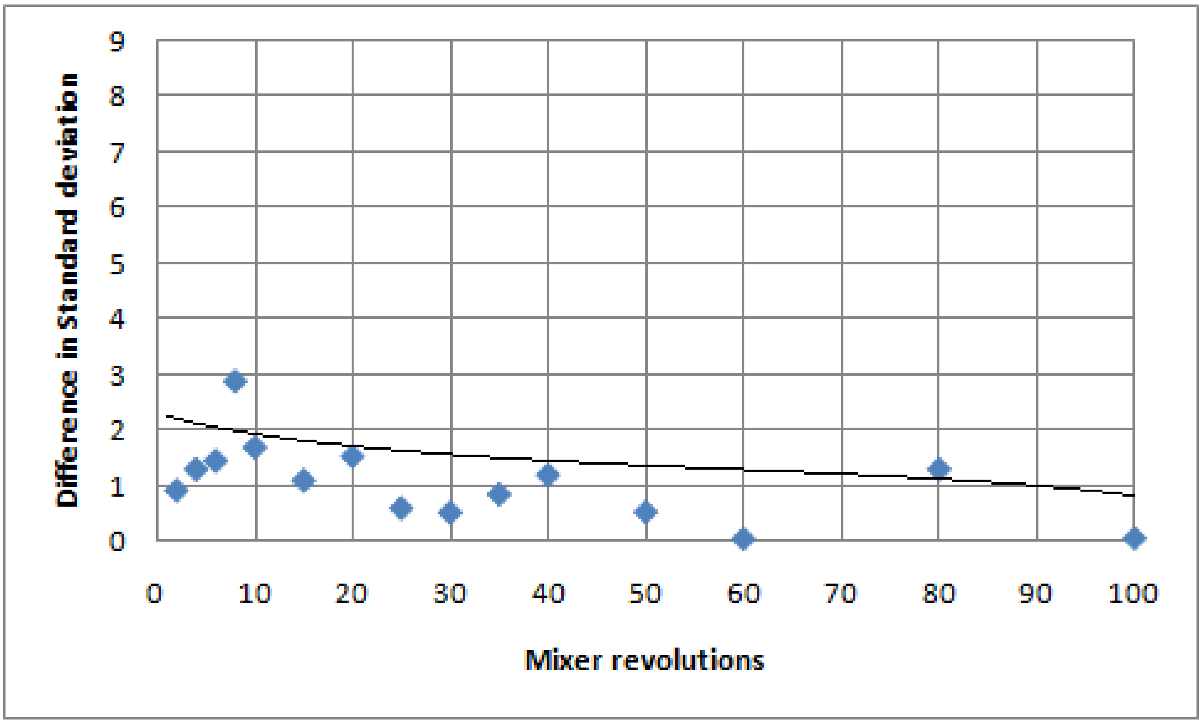

Figure 37: Difference in standard deviation between the two samples taken at every interval for the V-blender filled to 60% (Discussion of figure provided in text).

After reviewing Figure 34 to Figure 37, it was concluded that overall uniformity of the mixture was achieved when the difference in the standard deviation within the two selected samples was smaller than one.

For the Y-blender, this requirement was achieved within the first ten revolutions for both fill levels. The V-blender was also capable of achieving this level of uniformity within ten revolutions for a fill level of 30%, but not for a fill level of 60%.

Through these results, the same conclusion can be drawn as that following from the filtration procedure. This being that the observed Y-blender is less affected by the level to which it is filled than the observed V-blender.

3.4. Conclusion

The experimental results indicate that the evaluated Y-blender can produce a larger throughput of uniform material than the V-blender. This implies that the Y-blender is a distinct improvement over the V-blenders described in the literature. This is largely due to the improved mixing between the various arms of the Y-blender, whilst maintaining the effectiveness of the mixing within each of the arms individually.

Within the V-blenders, there is very little space at the base of the legs of the mixer for the material to be blended and separated as the mixer is rotated. This is illustrated in Figure 38.

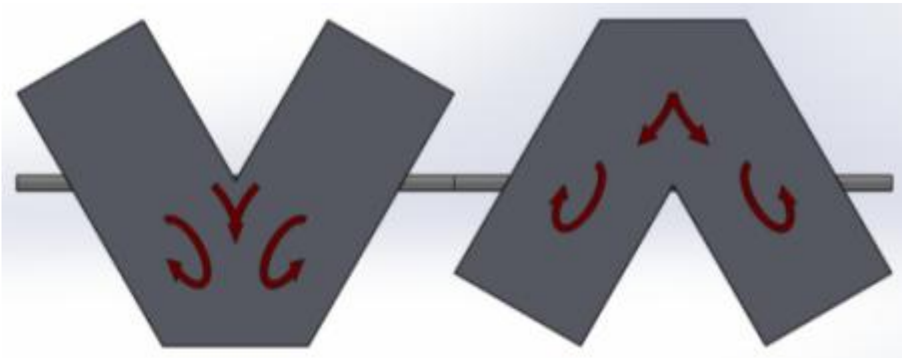

Figure 38: The areas where mixing generally occurs within a V-blender.

The added leg of the Y-blender provides additional space for the material to accumulate and mix, before it is separated once again between the other two legs as the mixer is rotated. This concept is illustrated in Figure 39.

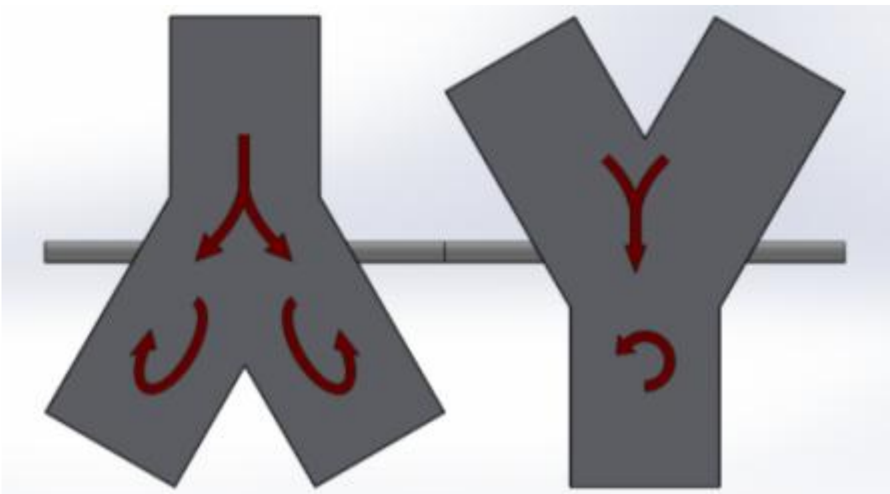

Figure 39: The areas where mixing can occur within the Y-mixer.

Chapter 4: Design

The layout of the mixing system is presented in this chapter, which leads to a description of how the various mixing system components are to be controlled by the control system. Once the concept of the control system is described, the design of the algorithm used by the control system to control the system components is then presented.

4.1. Mixing system layout

As discussed in Section 1.2 that describes the aim of the project, raw ingredients in milled form is stored in six large storage bins. An unload process will allow the material to be extracted from each bin in the correct proportion according to a predefined recipe. As the material is unloaded it is fed into a mixer by a tubular drag conveyor. The mixer is a Y-blender as described and evaluated in the previous chapter. After mixing the material for a defined period of time, the mixture is dumped into a bin from where it is fed to a pre-feeder bin where the material is stored until the feeder bin of the extruder requires more material. The level of the pre-feeder bin will initiate the pre-processing system.

Five recipes (presented in Appendix A) were used, as stated in the user specification requirements, to define the boundary values for the system design. When these recipes were compared, it was found that the recipe with the lowest density was the recipe for a maize-soy snack. The recipe was calculated to have an average density of 472.5kg/m^3. This material density was rounded to 450kg/m^3 and used throughout the sizing calculations of the mixing system.

By using these calculations the layout found in Figure 40 was generated to accomplish the required objectives.

By using this layout, for an average particle density of 450kg/m^3, it was found that the two feeder bins to the far left of Figure 40 have a capacity of 330kg at a fill level of 95%. The two bins in the middle then have capacity of 250kg at a fill level of 95% and the two bins on the right hand side in Figure 40, a capacity of 175kg at a fill level of 95%.

The Y-blender in this layout has a capacity of 50kg for a fill level of 66%. This implies that seven cycles will be required per hour for a throughput of 350kg/h, as required by the user specifications for this project.

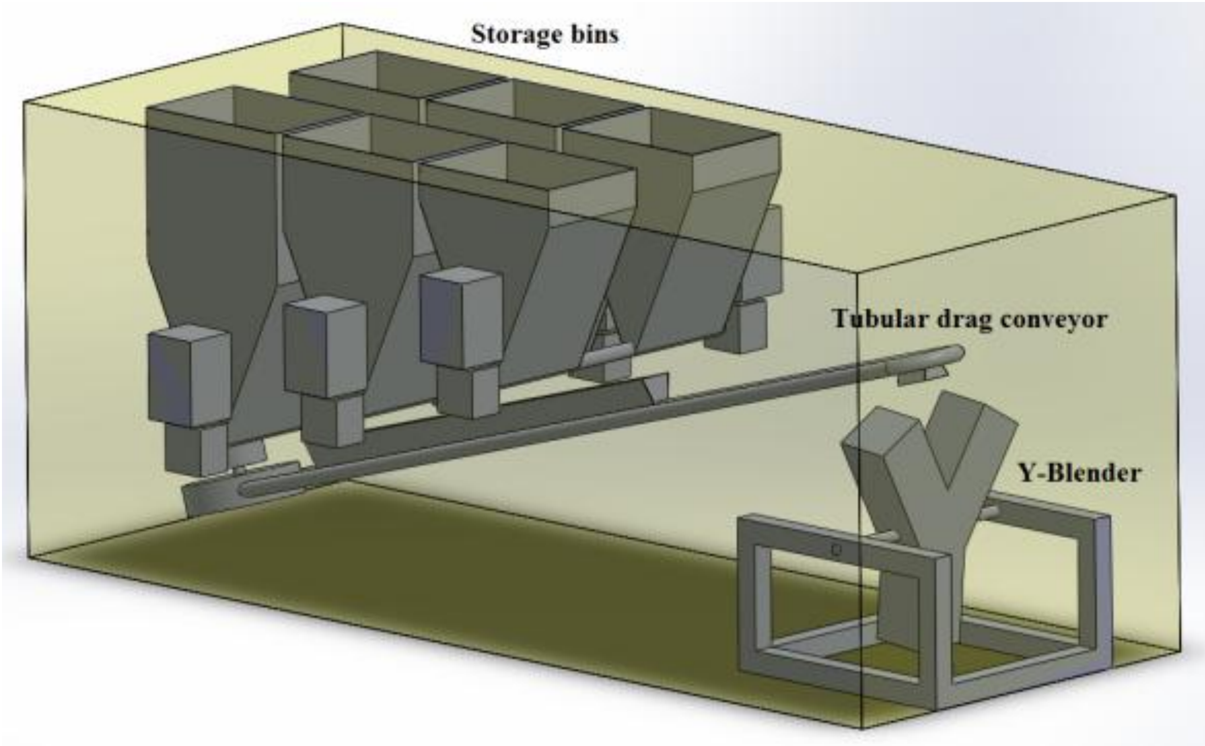

Figure 40: Layout of proposed mixing system within a container (Dimensions: L = 5900mm; H = 2400mm; W = 2350mm), showing six feeder bins capable of unloading material onto a tubular drag conveyor. The tubular drag conveyor subsequently feeds the material to a Y-blender.

4.2. Control layout

The previous section discussed the layout of the mixing system. This section will describe the concept by which each of the components within the layout is controlled.

Storage bins:

Each of the six storage bins is placed on a set of load cells, where each set of load cells is connected to a load cell junction box. These junction boxes produce six distinct analogue signals, representing the weight of each bin. These six analogue signals are fed to the controller as input values and are used for calculations relating to the feeding of material. The controller will manage the unloading of the bins by activating a relay coil for each bin.

Tubular drag conveyor:

The tubular drag conveyor is used to feed the material from the storage bins to the mixer. The conveyor will be started by the controller as soon as any of the storage bins starts feeding. The drag conveyor is a process where a fixed volume of material is transported at a certain rate. A timing system can therefore be used by the controller to stop the conveyor a specific time period after the last storage bin stopped feeding.

Mixer:

After the desired amount of material is extracted from each bin the controller will activate a relay, controlling the motor of the mixer and keep it activated for a defined period of time. Once the desired level of mix is reached the controller will activate another relay that controls a pneumatic cylinder. This pneumatic cylinder will unload the mixture from the mixer into the dump bin by opening a cover slide.

Dump bin:

The dump bin has a built in sensor that activates/deactivates the dump bin unload procedure. There is also a main control breaker to deactivate the capability of the automatic unloading of the dump bin, which is controlled by the controller.

Pre-feeder bin:

The pre-feeder bin is equipped with high/low sensors that provide the controller with signals representing the level of material within the bin. This value is then used by the controller to either initiate or stop the mixing process. Unloading of the pre-feeder bin is controlled by the level of the extruder feeder bin.

Human Machine Interface (HMI) and controller relationship:

The various recipes are stored on an HMI connected to the controller. A user can either select a pre-defined recipe from the HMI, or enter custom material values. A mixing time can also be specified, after which the mixing system will be initiated. The controller constantly sends new readings from the input signals to the HMI, which responds with actions for the controller to perform. The HMI therefore forms a critical part of the control process.

A summary of this control layout concept is found in Figure 41.

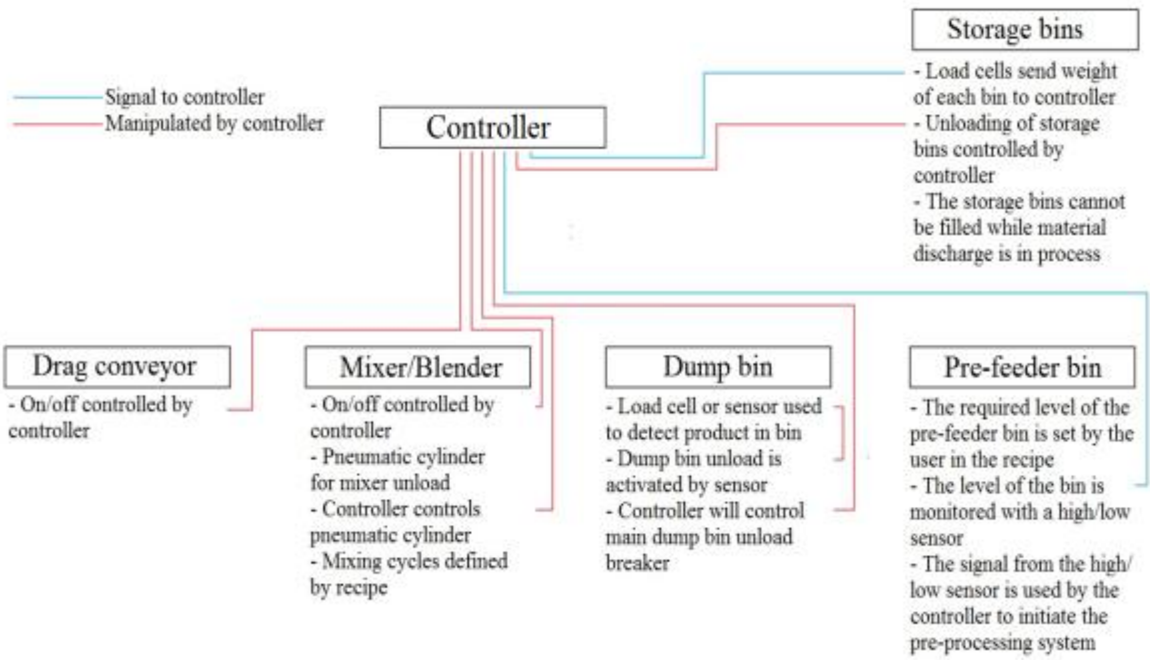

Figure 41: Concept of mixing system control.

4.3. Description of control algorithm design

The control system was designed to operate on five levels. The first level is just a menu page that enables the user to choose between manual and automatic operation.

If the manual option is chosen the user is presented with the second level of the control program. In this second level all the current weights of the storage bins are displayed on the screen and the user can touch the icons on the screen to control the storage bins, chain conveyor, mixer and mixer door.

If automatic operation is chosen on the first level the user is directed to a third level containing another menu page. Here the user can choose to load an existing recipe or to create a new recipe.

If the user chooses to create a new recipe he/she is directed to a fourth level where a new recipe can be entered. Upon saving the new recipe, the user is redirected back to the third level.

The user is directed to the fifth level of the control program upon choosing and loading a recipe. This level starts the automatic mixing process. Within this level the status of the mixer, chain conveyor and mixer door are displayed on the screen along with the weights of the storage bins. The user can therefore observe the weighing and mixing processes.

The fifth level is responsible for automatically weighing material as it is fed from the storage bins, feeding that material to the mixer, and mixing it. All of this is done according to pre-determined recipes.

The main control loop within the fifth level is the most critical control section and will therefore be the only part discussed further within the text. The other sections of the control program can be found in the appendices, but as it is simple to realize the operation of the code, the explanation of the program is left to the imagination of the reader.

The design of the main control loop within the fifth level is simplified and presented in Figure 42 and its function is explained below.

The main control loop refers to a series of code that is repeated over and over for as long as it is active. This loop is called after the chosen recipe is loaded and the automatic weighing and mixing process is initiated. This loop will continue running on the HMI, as long as the user remains on the fifth level and the HMI is connected to the controller.

The control loop first verifies whether the “on” or “off” buttons are pressed to determine whether or not to initiate the system. Once the *on* button is pressed, the weighing and mixing parts of the control program is initiated and will stay initiated until the *off* button is pressed.

If the loop observes that the “on” button is active it is first paused for 0.3 seconds. This is done to slow down the readings on the load cells and allow for some material to be extracted from the storage bins if required. If this delay is shortened, the sampling rate of the readings will increase, and the general response time of the control program will decrease. This does however require more memory from the HMI and it is also limited by the data transfer rates between the HMI and the controller, as well as the data transfer rates between the controller and the other peripherals.

The loop then verifies if it should start the weighing or the mixing process. If the weighing process is to be carried out, the loop needs to determine whether or not it needs to calculate a new final bin weights, or set points, for the current batch. This is done by setting and resetting a log parameter to indicate to the loop what processes were executed during the last repetition of the loop.

If the loop needs to log a new weight, the program will read the current weight of the storage bin, subtract the amount required by the recipe and log this as set points to the loop. The loop will then also create a log entry to inform itself that the reading was carried out. This entry will be used during the following repetition of the loop to state that the set point was created.

The loop will then ensure that the mixer is not in operation and that the mixer door is open, before starting the chain conveyor. After the chain conveyor is started, the loop will continue to the weighing/feeding process.

A series of logic tests are presented to the loop, but only the first one that is valid will be carried out. The objective of the logic tests is to determine whether or not the set point for the weight of the storage bin was reached or not.

Once the current weight no longer exceeds the goal weight, the loop will deactivate the bin discharge motor and then pause for three seconds before de-activating the chain conveyor and closing the mixer door. The delay is to ensure that all the material in the conveyor has the required time to be fed into the mixer.

The loop will log a parameter to indicate that the weighed feeding process should come to an end and that, after the mixing process, new goal set points should be established. The loop will then initiate the mixing process.

During the mixing process the loop will once again ensure that the chain conveyor is de-activated and that the mixer door is closed. The loop then reads the appropriate mixing time from the recipe and activates the mixer. After the mixer is activated, the loop will be paused for the period of time set in the recipe. The mixer is de-activated once the time expires. The loop then opens the mixer door to eject the material from the mixer. The door is then closed after a short delay. The mixer is activated, a short delay is used and then the mixer is de-activated again. This is done to turn the mixer so that it faces the right way to be filled during the next repetition.

The mixer door is opened again and the loop creates a log entry to show that the following repetition should initiate the weighing process. This marks the end of the loop while the “on” button is pressed.

The loop will ensure that the mixer, storage bins and chain conveyor is de-activated and that the mixer door is open if the “off” button is pressed. The loop will then initiate another 0.3 second delay before starting the next repetition.

- The system checks if the on button is pressed
  - The loop is paused for 0.3 seconds
  - The system checks if it should start the weighing process
    - The system checks if it should log a new bin weight
      - The system reads the weight of the first bin
      - The system reads the amount of material required by the recipe
      - The system does the necessary calculations and saves a setpoint for the bin
      - The system logs that the weight have been set for the bin
    - The system ensures that the mixer is not in operation and the mixer door is open
    - The system activates the chain conveyor
    - The system then completes the first of the following options that is valid
      - The system checks if the weight in the bin is higher than the setpoint
        - The system activates the bin discharge motor
      - The system checks if the weight in the bin is not higher than the setpoint
        - The system de-activates the bin discharge motor
        - The loop is paused for 3 seconds
        - The system de-activates the chain conveyor
        - The system closes the mixer door
        - The loop is paused for 1.5 seconds
        - The system logs that a new weight setpoint should be established
        - The system logs that the a batch was weighed out
  - The system checks if the weighing process is completed
    - The system ensures that the chain conveyor is not in operation and the mixer door is closed
    - The system activates the mixer
    - The system reads the mixing time required by the recipe
    - The loop is paused for the required time period for mixing
    - The system de-activates the mixer
    - The loop is paused for 1.5 seconds
    - The system opens the mixer door for material discharge
    - The loop is paused for 3 seconds
    - The system closes the mixer door
    - The loop is paused for 3 seconds
    - The system activates the mixer to turn it upright
    - The loop is paused for 3 seconds
    - The system opens the mixer door
    - The loop is paused for 3 seconds
    - The system logs that the weighing process should be restarted
- The system checks if the off button is pressed
  - The system ensures that the mixer, chain conveyor and bin motors are switched of
  - The system ensures that the mixer door is open
  - The loop is paused for 0.3 seconds

Figure 42: A flow diagram representing the fifth level of the control program as discussed within the text.

Chapter 5: Evaluation of the system control

In this chapter the effectiveness of the designed mixing control system is evaluated. This is first done by presenting the detail design of the control hardware, followed by the implementation of the control system and conclusions regarding the feasibility of the control system.

5.1. Control hardware

The control subsystem consists of an IOIO USB controller produced by SparkFun Electronics. The IOIO USB controller forms a bridge between a device running on the "Android" operating system, like a tablet or cell phone, and the physical components of the mixing system. The controller can be connected to the tablet or cell phone via USB or Bluetooth and an Android application, such as the one presented in the previous chapter, is used to control the system components through the controller.

The IOIO USB controller weighs 11g, has dimensions of 77mm X 32mm X 5mm and consists of 48 General Purpose I/O (GPIO) pins that can be used to connect 3.3V digital inputs or outputs. Of these 48 GPIO pins, 16 can also be used as analogue inputs or as analogue outputs by using Pulse Width Modulation (PWM). A PWM output is where the duty cycle (that is the ratio between the pulse width and the pulse period) of the signal is increased or decreased in order to approximate analogue signals. Some of the GPIO pins can also perform other functions, such as additional data module connections, but these functions were not used in this study and will therefore not be discussed further.

An image of an IOIO USB controller is presented in Figure 43.

Figure 43: An IOIO USB microcontroller (As discussed above).

To evaluate the feasibility of using an IOIO USB microcontroller as a possible control solution, a scale model of the crucial components of the mixing system was designed and built. The control system was also manufactured similar to the configuration for the full scale system.

The control system was then used to operate the model system to evaluate the feasibility of using it on the full scale system.

5.2. Control system setup

The circuit diagram for the complete mixing system can be found in Figure 44. The functions of the highlighted and labelled parts of the circuit within the figure will be discussed individually.

Since the control of the feeder bins rely on similar input and output signals, the control of the six feeder bins is identical. Accordingly, the control of only one feeder bin will be used to describe and illustrate the working concept.

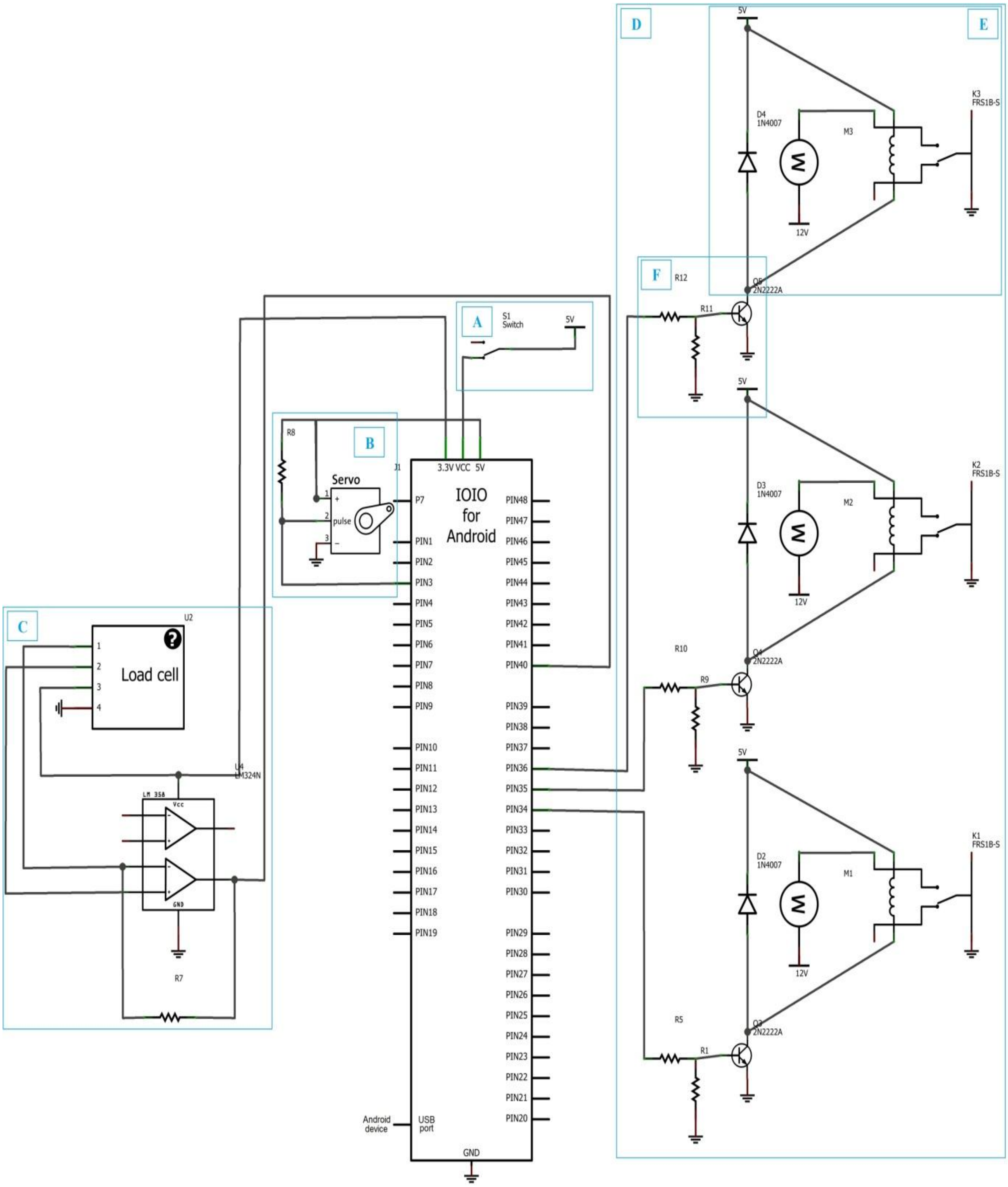

Figure 44: The circuit diagram of the control system used for the scale model mixing system.

The first circuit, Part A of the circuit diagram in Figure 44, shows the master switch, required to activate or deactivate the entire mixing system.

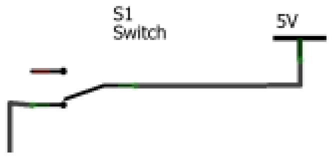

Figure 45: Part A – shows the master switch used to activate or deactivate the entire control system.

The second circuit, Part B of the circuit diagram in Figure 44, highlights the servo control system. Only a relatively small servo motor is required for the model system, and it was therefore connected directly to the IOIO controller with a 10kΩ resistor connecting the sense and excite pins with an increased 5V input. This method can also be used as an analogue output to control variable speed drivers for larger motors. The layout of this type of connection is shown in the circuit diagram in Figure 46.

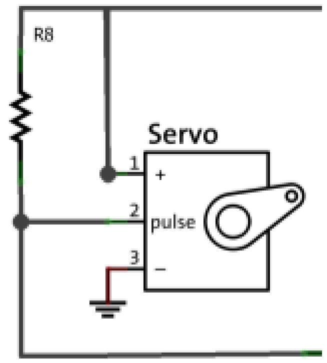

Figure 46: Part B – Circuit diagram used to connect the servo motor to the control system.

The third circuit, Part C of the circuit diagram in Figure 44, highlights the part of the control system required to receive a signal from the load-cell. The scale model mixing system only required a single load-cell to be sensible due to the size of the feeder. A LM358, LM324N or similar Op-amp is used in this design to amplify the signal from the load cell to ensure that a correct reading is received by the controller. In this design a 100kΩ resistor is used to scale the signal from the Op-amp. The circuit diagram for this connection is presented in Figure 47.

As stated, the scale model system only uses one load-cell per feeder, whereas this is not the common practice in the industry. When it is not feasible to balance the feeder on a single load cell and multiple load cells are required, a junction box, like the LT540-5 way IP65 junction box, is used to convert the signals from multiple load-cells to a single signal. An output is then received from the junction box and sent to the controller. The junction box is connected to the controller in the same manner as described earlier to connect the single load-cell.

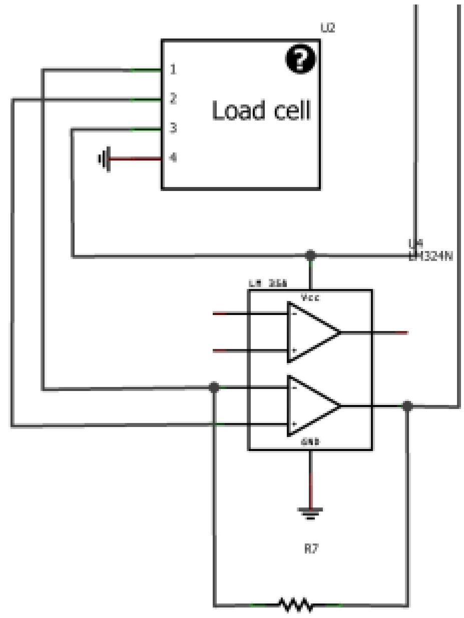

Figure 47: Part C – Circuit diagram of the components used to connect the load-cells to the control system.

The fourth circuit, Part D of the circuit diagram in Figure 44, highlights the part of the control system used to control one of the feeder motors, the motor of the chain conveyor as well as the motor driving the mixer. The layouts of the part of the control system controlling these three motors are very similar as can be seen in Figure 44. In each case, the motor is controlled by a relay system as highlighted in Part E of the circuit diagram in Figure 44. An enlarged version of this part of the circuit can also be seen in Figure 48.

For the model system the relays used only had a capacity of 24W. To use this control system on a full scale mixing system the relays should be replaced with relays of the appropriate capacity.

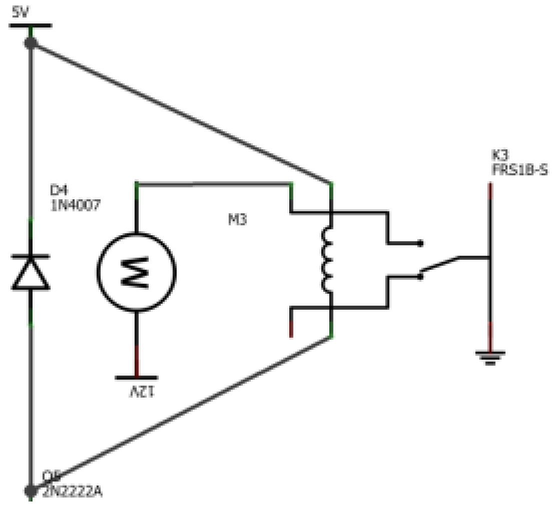

Figure 48: Part D – Circuit diagram of the components used to drive the various motors of the mixing system components.

The IOIO controller only provides 3.3V output on the selected pins, whereas the relays required a switching voltage of at least 5V. A circuit therefore needed to be introduced to increase the output voltage from the IOIO controller. This circuit is highlighted in Part E of the circuit diagram in Figure

44, where a 2N2222A transistor is used together with a 1N4007 diode, a 1kΩ resistor and a 330Ω resistor. This circuit provides a 5V pulse at the relay coil when a 3.3V pulse is received over the 330Ω resistor. An enlarged image of this part of the circuit diagram can be found in Figure 49.

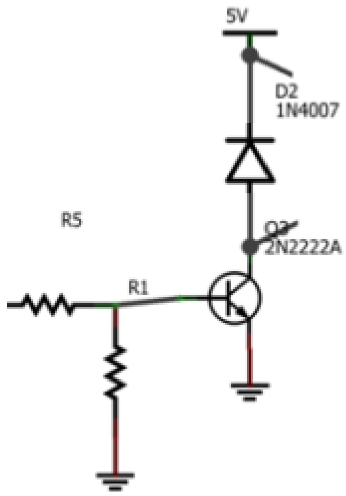

Figure 49: Part E – Circuit diagram of the components used to increase the voltage from the output of the IOIO controller from 3.3V to 5V.

5.3. Control system implementation

As stated earlier, a model of the key components of the mixing system was manufactured in order to show that the controller design is applicable to a commercial mixing system. The control circuits required to manage these components were also produced as described in the previous section, along with the required software for the user interface.

A system of breadboards was utilized to connect the control circuitry to the IOIO USB controller for the evaluation procedures. This allowed for changes to be made to the configuration of the circuitry as the study developed, without incurring additional expenses during each change. The layout of the breadboard is given in Figure 50 below. A photograph of the final breadboard configuration as used for the evaluation of the control system is added in the bottom right corner of Figure 50.

In the layout presented in Figure 50, the three motors represent the motors of the mixer, one of the storage bin feeders and the chain conveyor. The servo motor in Figure 50 represents the servo motor used to control the mixer door. The load-cell in the figure represents the signal from the load-cell used to calculate the weight of the feeder bin.

All of the components within the model of the mixing system were controlled in the same manner as the large scale setup would be controlled, the only difference being the capacity of the relays and motor drives. If these components were to be switched with the capacity required for the larger setup, the same control system could be used to control and operate the large setup.

The cost of a control system using a PLC to control the same mixing system components as controlled by the IOIO USB controller, would be much higher and the size of said control system would also be larger than the size of the suggested control system.

Changes and improvements to any system are inevitable within a research environment. A low cost control system where changes can easily be made without incurring a lot of additional charges would therefore be a very practical and attractive solution. The IOIO USB controller was able to accomplish this during the evaluation procedure.

It can therefore be concluded that the suggested control system is an affordable control solution of which the configuration is easy to upgrade or reconfigure. The control system was also fully capable of managing all the required components, making it feasible for this study.

A video of the model system being controlled by the described control system can be found directly on: http://youtu.be/LlGaN3GXkZk or by searching for "DJ Kruger IOIO USB Controlled Mixing System" on YouTube.

As stated earlier the layout of the control circuitry on the breadboards as used to evaluate the control system is presented in Figure 50.

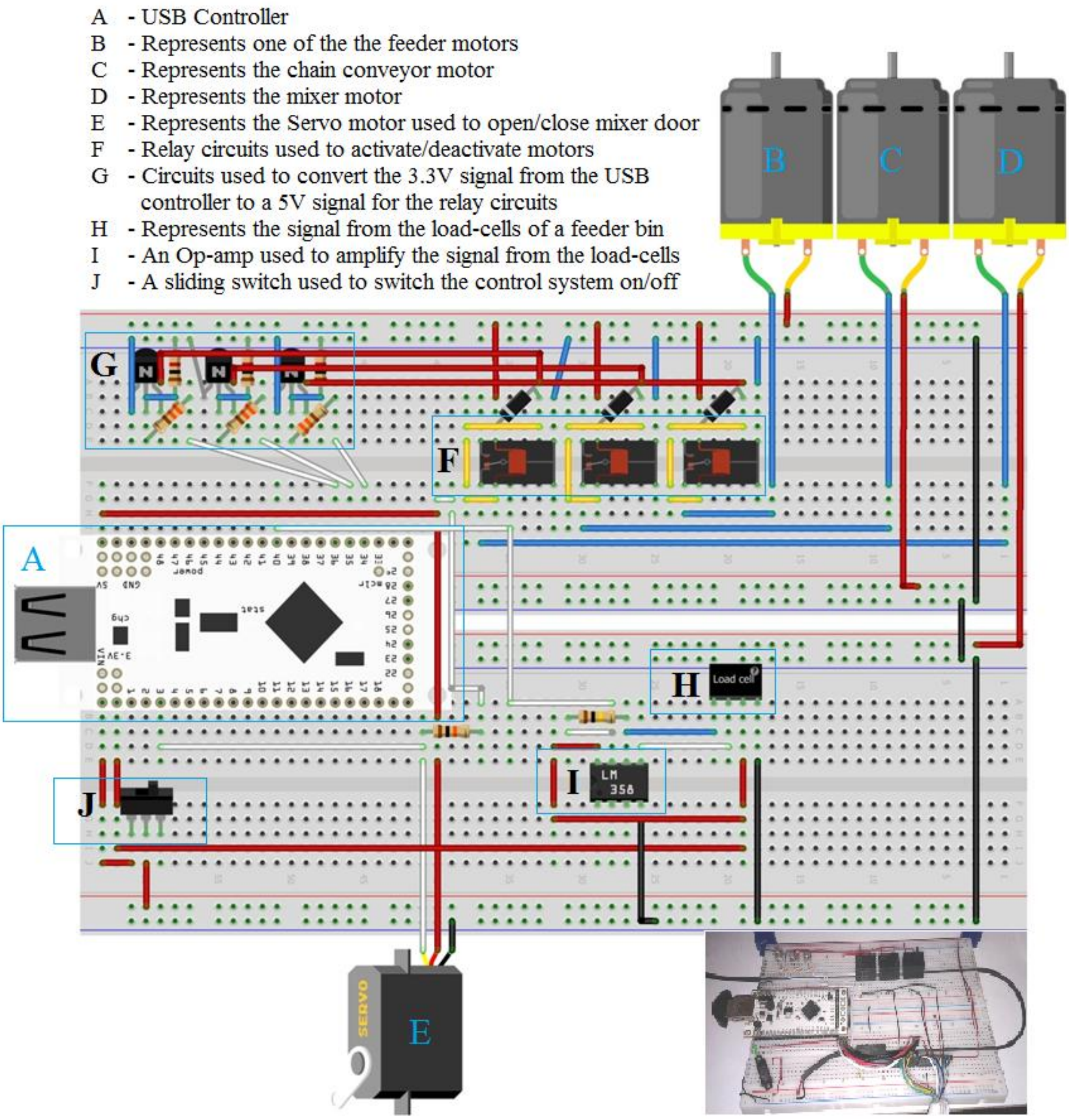

Figure 50: The layout of the breadboard as used to control the scaled mixing system.

Chapter 6: Conclusion

A discussion of the conclusions of this study is presented in this chapter as well as recommendations for further studies or investigations that can sprout from these findings.

6.1. Discussion and conclusions

The aim of this study was to develop a setup that would enable clients to optimise their mixing and lead studies in the field of extrusion pre-processing. This setup also needed to facilitate demonstrations and training and therefore the setup had very specific size restrictions. The costs of the setup also lead to certain cost restrictions.

During this study mixing was identified as one of the key processes during pre-processing and therefore identified as an area that could be developed.

A V-blender was identified as a possible mixing solution. Cleary & Sinnott (2008) suggested a change within the symmetry of the mixer to achieve better mixing results. After reviewing the literature, it was hypothesized that adding a third leg to the V-blender could also improve the efficiency of the V-blender. By adding a third leg, a Y-blender was put forward.

This suggested Y-blender was therefore incorporated into the design of the setup. Models of the Y-blender as well as the V-blender described within the literature were then manufactured and used to evaluate and characterize the effectiveness of a Y-blender.

It was found that a Y-blender could achieve much better mixing of large quantities of granular material than similar mixers found in the literature. This suggested that a smaller mixer could be incorporated within the setup to deliver the same throughput as its larger counterpart.

It was found that, by using the layout set forth in Figure 40, the size restrictions and the mixing requirements of the project could be met.

As stated in the layout design, for an average particle density of 450kg/m^3 and a fill level of 95%, the system will contain two bins with 330kg capacity each, two bins with 250kg capacity each and two bins with 175kg capacity each. The suggested Y-blender will have a capacity of 50kg at a fill level of 66%, requiring only seven cycles per hour to deliver the required throughput of 350kg/h.

The study indicated that this was a realistic expectation for this mixing system, and the proposed layout of the setup within a container (Dimensions: L = 5900mm; H = 2400mm; W = 2350mm) was considered viable.

As part of the aim of the study, a control system was required that would be both inexpensive and accurate.

The IOIO USB controller is an affordable control solution. The configuration of this control system is also easy to upgrade or reconfigure without causing a lot of additional costs, making it an ideal candidate to address the requirements of this study within a research environment.

When compared to a PLC controller, which is typically used in the industry, an IOIO USB microcontroller is less expensive and will occupy less space than a PLC with the same capabilities.

This indicated that a considerably lower cost control system could be implemented on the setup, greatly reducing the overall cost of the design considered.

This project identified a Y-blender to be an ideal mixing solution for a portable particle mixing facility and also showed how an IOIO USB controller can be implemented as an affordable control solution.

6.2. Recommendations

A possible further improvement can be made to the Y-blender by placing a triangular blade around the mixer shaft where it passes through the blender.

During this experiment, a tubular shaft was used to rotate the blender. This caused the material to flow around the shaft and caused very little distortion within the particle flow. If a triangular blade was to be used, it would be possible that further manipulation of the particle flow can be achieved, thus increasing the effectiveness of the mixer.

Figure 51 shows the tubular shaft used to rotate the experimental Y-blender.

Figure 51: The tubular shaft used to rotate the experimental Y-blender.

6.3. Further investigations/future studies

A suggestion is that further studies can be carried out on the mixer, where the full scale Y-blender can be simulated using DEM, as described in Section 2.3, to analyse the flow of the material within the mixer.

A positron camera can also be used to track a particle within the Y-blender for a given number of rotations using PEPT, also described in Section 2.3. This can also be done on a full scale Y-blender to validate the DEM simulations.

If a colorant that is soluble in water is used, similar to the one used in this study during the filtration procedure, the colour of the liquid after filtration can also be used to evaluate the rate that mixing occurs within the mixer. To do this, each sample must be weighed and an exact percentage of water must then be passed through the filter, for instance a hundred millilitres of water for every gram of sample material. The clarity of the water will then represent the amount of colorant that was present within the sample.

Bibliography

Aissa, A., Duchesne, C. & Rodrigue, D., 2011. Effect of friction coefficient and density on mixing particles in the rolling regime. *Powder Technology*, 212(2), pp.340-47.

Borzenski, F.J., 1999. An overview of variables affecting batch mixing in a tangential mixer. *Rubber world*, 219(6), pp.24-32.

Bridgwater, J., 2003. The dynamics of granular materials - towards grasping the fundamentals. *Granular Matter*, 4(4), pp.175-81.

Cleary, P.W. & Sinnott, M.D., 2008. Assessing mixing characteristics of particle-mixing and granulation devices. *Particuology*, 6(6), pp.419-44.

Finnie, GJ; Kruyt, NP; Ye, M; Zeilstra, C; Kuipers, JAM, 2005. Longitudinal and transverse mixing in rotary kilns: A discrete element method approach. *Chemical Engineering Science*, 60(15), pp.4083-91.

Ingram, A; Seville, JPK; Parker, DJ; Fan, X; Forster, RG, 2005. Axial and radial dispersion in rolling mode rotating drums. *Powder Technology*, 158(1-3), pp.76-91.

Kalpakjian, S. & Schmid, S.R., 2010. *Manufacturing engineering and technology, 6th edition in SI units*. Singapore: Prentice Hall.

Khakhar, D., McCarthy, J., Gilchrist, J. & Ottino, J., 1999. Chaotic mixing of granular materials in two-dimensional tumbling mixers. *Chaos*, 9(1), pp.195-205.

Kohlgrüber, K., 2008. *Co-Rotating Twin-Screw Extruders, Fundamentals, technology and applications*. Ohio: Hanser Gardner Publications, Inc.

Kuo, HP; Knight, PC; Parker, DJ; Tsuji, Y; Adams, MJ; Seville, JPK, 2002. The influence of DEM simulation parameters on the particle behaviour in a V-mixer. *Chemical Engineering Science*, 57(17), pp.3621-38.

Laurent, B. & Bridgewater, J., 2002. Performance of single and six-bladed powder mixers. *Chemical Engineering Science*, 57(10), pp.1695-709.

Laurent, B., Bridgewater, J. & Parker, D., 2004. Motion in a Particle Bed Agitated by a. *AIChE Journal*, 46(9), pp.1723-34.

Metcalfe, G., Cleary, P.W. & Liffman, K., 1998. How well do discrete element granular flow models capture the essentials of mixing processes? *Applied Mathematical Modelling*, 22(12), pp.995-1008.

Mohamed, S., 1990. Factors affecting extrusion characteristics of expanded starch-based products. *Journal of Food Processing and Preservation*, 14(6), pp.437-52.

Ottino, J. & Khakhar, D., 2000. Mixing and Segregation of Granular Materials. *Annual Review of Fluid Mechanics*, 32(1), pp.55-92.

Remy, B., Glasser, B.J. & Khinast, J.G., 2010. The effect of mixer properties and fill level on granular flow in a bladed mixer. *AIChE Journal*, 56(2), pp.336-53.

Riaz, M., 2011. Introduction to Extrusion. In *Aquaculture Feed Extrusion, Nutrition, & Feed Management Practical Short Course, 18th ed*. College Station, 2011.

Sarkar, A. & Wassgren, C.R., 2009. Simulation of a continuous granular mixer: Effect of operating conditions on flow and mixing. *Chemical Engineering Science*, 64(11), pp.2672-82.

Sarkar, A. & Wassgren, C., 2010. Continuous blending of cohesive granular material. *Chemical Engineering Science*, 65(21), pp.5687-98.

Smith, P.G., 2011. *Introduction to Food Process Engineering*. Second Edition ed. New York: Springer Science+Business Media, LLC.

Stewart, R., Bridgewater, J., Zhou, Y. & Yu, A., 2001. Simulated and measured flow of granules in a bladed mixer—a detailed comparison. *Chemical Engineering Science*, 56(19), pp.5457-71.

Wang, L., Smith, S. & Chessari, C., 2008. Continuous-time model predictive control of food extruder. *Control Engineering Practice*, 16(10), pp.1173 - 1183.

Appendix A

Appendix A contains recipe and material information used during the calculations for this study. Five recipes were considered consisting of ten different materials (Four of the ten materials were added manually to the system to reduce the size of the system). Table 7 shows the composition of each recipe and Table 8 shows the average density of each recipe. These densities were used during the calculations for the mixing systems.

Recipe 1: Maize soy cereal

Recipe 2: Maize soy snack

Recipe 3: Multi-grain breakfast cereal

Recipe 4: Wheat soy cereal

Recipe 5: Maize sugar snack

Table 7: Recipe composition – The ratio (%) of materials required for each recipe (Recipe 1 – Maize soy cereal: 63.8% Super maize meal, 21.3% De-fatted soy flour, 13.5% Sugar, 1.5% Salt).

Material		Recipe 1	Recipe 2	Recipe 3	Recipe 4	Recipe 5
Storage bin 1:	Super maize meal	63.8	52.5	29.4	0	86.2
Storage bin 2:	De-fatted soy flour	21.3	17.5	22.5	23.4	0
Storage bin 3:	Wheat flour	0	0	17.3	47.7	0
Storage bin 4:	Rolled oats	0	0	17.3	0	0
Storage bin 5:	Wheat germ	0	0	0	18.9	0
Storage bin 6:	Sugar	13.5	0	11.1	9	13.8
Manual 1:	Salt	1.5	1.8	1.23	1	0
Manual 2:	Flavourant	0	7.2	0.62	0	0
Manual 3:	Vitamin mixture	0	0	0.62	0	0
Manual 4:	Sunflower oil	0	21	0	0	0

Table 8: The average density of each material and the various recipes (Recipe 5 – Maize sugar snack @ +/- 715kg/m^3: 86.2% Super maize meal @ 700kg/m^3, 13.8% Sugar @ 800kg/m^3).

Material densities (kg/m^3)		Recipe 1	Recipe 2	Recipe 3	Recipe 4	Recipe 5
S1	700	446.25	367.5	205.695	0	603.4
S2	600	127.5	105	134.82	140.4	0
S3	750	0	0	129.6375	357.75	0
S4	300	0	0	51.855	0	0
S5	650	0	0	0	122.85	0
S6	800	108	0	88.888	72	110.4
M1	1200	18	21.6	14.814	12	0
M2	600	0	43.2	3.708	0	0
M3	600	0	0	3.708	0	0
M4	920	0	193.2	0	0	0
Average density of recipe		699.75	730.5	633.1255	705	713.8

Appendix B – Complete .xml and .java code for control application

Main menu .xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/particlespic" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="@string/welcometext"
        android:textSize="@dimen/welcomescreentext" />

    <Button
        android:id="@+id/bmanual1"
        android:layout_width="wrap_content"
        android:layout_height="140dp"
        android:layout_below="@+id/textView1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"
        android:text="@string/bmanual"
        android:textSize="@dimen/welcomescreentext" />

    <Button
        android:id="@+id/bauto1"
        android:layout_width="wrap_content"
        android:layout_height="140dp"
        android:layout_below="@+id/bmanual1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"
        android:text="@string/bauto"
        android:textSize="@dimen/welcomescreentext" />

</RelativeLayout>
```

Main menu .java

```
package com.mixer.control;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {

  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
```

```
        Button man1 = (Button) findViewById(R.id.bmanual1);
        man1.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                                // TODO Auto-generated method stub
                                startActivity(new Intent("com.mixer.control.MANUALONE"));
                        }
                });
        Button auto1 = (Button) findViewById(R.id.bauto1);
                auto1.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                                // TODO Auto-generated method stub
                                startActivity(new Intent("com.mixer.control.AUTOONE"));
                        }
                });
    }
        @Override
        protected void onPause() {
                // TODO Auto-generated method stub
                super.onPause();
        }
}
```

Manual operation .xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/particlespic"
    android:orientation="vertical" >

 <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/twmancurrent"
        android:textSize="@dimen/manualscreentext" />

     <TextView
         android:id="@+id/textView7"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_marginLeft="@dimen/chainmarman"
         android:text="@string/tchainman"
```

```
            android:textSize="@dimen/manualscreentext" />

</LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

    <TextView
        android:id="@+id/mfb1weight"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:text="@string/twmanualbinweight"
        android:textSize="@dimen/manualscreentext" />

    <ToggleButton
        android:id="@+id/tbBin1"
        android:layout_width="200dp"
        android:layout_height="70dp"
        android:background="@drawable/binpic"
        android:padding="@dimen/manpadding"
        android:text="@string/tbman1"
        android:textSize="@dimen/manualscreentext" />

     <ToggleButton
         android:id="@+id/tbChainconv"
         android:layout_width="200dp"
         android:layout_height="70dp"
         android:layout_marginLeft="@dimen/chainmarmanb"
         android:background="@drawable/chianconvpic"
         android:padding="@dimen/manpadding"
         android:text="@string/tbman1"
         android:textSize="@dimen/manualscreentext" />

</LinearLayout>
<LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

    <TextView
        android:id="@+id/textView3"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:text="@string/twmanualbinweight"
        android:textSize="@dimen/manualscreentext" />

    <ToggleButton
        android:id="@+id/tbBin2"
        android:layout_width="200dp"
        android:layout_height="70dp"
        android:background="@drawable/binpic"
        android:padding="@dimen/manpadding"
        android:text="@string/tbman2"
        android:textSize="@dimen/manualscreentext" />

     <TextView
         android:id="@+id/textView8"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="@dimen/chainmarmanb"
        android:text="@string/manmix"
        android:textSize="@dimen/manualscreentext" />

</LinearLayout>
<LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

    <TextView
        android:id="@+id/textView4"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:text="@string/twmanualbinweight"
        android:textSize="@dimen/manualscreentext" />

    <ToggleButton
        android:id="@+id/tbBin3"
        android:layout_width="200dp"
        android:layout_height="70dp"
        android:background="@drawable/binpic"
        android:padding="@dimen/manpadding"
        android:text="@string/tbman3"
        android:textSize="@dimen/manualscreentext" />

    <ToggleButton
        android:id="@+id/tbMix"
        android:layout_width="200dp"
        android:layout_height="70dp"
        android:layout_marginLeft="@dimen/chainmarmanb"
        android:background="@drawable/mixerpic"
        android:padding="@dimen/manpadding"
        android:text="@string/tbman1"
        android:textSize="@dimen/manualscreentext" />

</LinearLayout>
<LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

     <TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="@dimen/manmixdoorhintpos"
        android:text="@string/manmixdoor"
        android:textSize="@dimen/manualscreentext" />

</LinearLayout>
<LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

    <Button
```

```
        android:id="@+id/mbopen"
        android:layout_width="190dp"
        android:layout_height="70dp"
        android:layout_marginLeft="@dimen/mandoorpos"
        android:background="@drawable/mixdoorpic"
        android:padding="@dimen/manpadding"
        android:text="@string/manmixopen"
        android:textSize="@dimen/manualscreentext" />

    <Button
        android:id="@+id/mbclose"
        android:layout_width="190dp"
        android:layout_height="70dp"
        android:background="@drawable/mixdoorpicclose"
        android:padding="@dimen/manpadding"
        android:text="@string/manmixclose"
        android:textSize="@dimen/manualscreentext" />

</LinearLayout>

</LinearLayout>
```

Manual .java

```
package com.mixer.control;

import ioio.lib.api.AnalogInput;
import ioio.lib.api.DigitalOutput;
import ioio.lib.api.PwmOutput;
import ioio.lib.api.exception.ConnectionLostException;
import ioio.lib.util.BaseIOIOLooper;
import ioio.lib.util.IOIOLooper;
import ioio.lib.util.android.IOIOActivity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
import android.widget.ToggleButton;

public class ManualOne extends IOIOActivity implements OnClickListener {
	@Override
	protected void onPause() {
		// TODO Auto-generated method stub
		super.onPause();
	}
	@Override
	protected void onStop() {
		// TODO Auto-generated method stub
		super.onStop();
	}
```

```
private ToggleButton mfb1, mchainconv, mmix;
private Button mdooropen, mdoorclose;
private TextView mwfb1;
private boolean doorpos = true;

@Override
protected void onCreate(Bundle savedInstanceState) {
	// TODO Auto-generated method stub
	super.onCreate(savedInstanceState);
	setContentView(R.layout.manual1);
	mfb1 = (ToggleButton) findViewById(R.id.tbBin1);
	mchainconv = (ToggleButton) findViewById(R.id.tbChainconv);
	mmix = (ToggleButton) findViewById(R.id.tbMix);
	mwfb1 = (TextView) findViewById(R.id.mfb1weight);
	mdooropen = (Button) findViewById(R.id.mbopen);
	mdooropen.setOnClickListener(this);
	mdoorclose = (Button) findViewById(R.id.mbclose);
	mdoorclose.setOnClickListener(this);
}

class Looper extends BaseIOIOLooper {
	private DigitalOutput FB1_PIN;
	private DigitalOutput CHAINCONV_PIN;
	private DigitalOutput MIX_PIN;
	private AnalogInput FB1W_PIN;
	private PwmOutput MIXDOOR_PIN;

	@Override
	protected void setup() throws ConnectionLostException, InterruptedException {
		FB1_PIN = ioio_.openDigitalOutput(34, true);
		CHAINCONV_PIN = ioio_.openDigitalOutput(35, true);
		MIX_PIN = ioio_.openDigitalOutput(36, true);
		FB1W_PIN = ioio_.openAnalogInput(40);
		MIXDOOR_PIN = ioio_.openPwmOutput(3, 100);
	}

	@Override
	public void loop() throws ConnectionLostException, InterruptedException {
		//Dirk -- jy moet die lyn net onder die try bokant hom paste
		FB1_PIN.write(mfb1.isChecked());
		CHAINCONV_PIN.write(mchainconv.isChecked());
		MIX_PIN.write(mmix.isChecked());
		if (doorpos == true)
			MIXDOOR_PIN.setPulseWidth(920);
		else
```

```
					MIXDOOR_PIN.setPulseWidth(1400);
				try {
					Thread.sleep(300);
				} catch (InterruptedException e) {
				}
				try {
					final float reading1 = FB1W_PIN.read();
					setText1(Float.toString((reading1*100)));

				} catch (InterruptedException e) {
					ioio_.disconnect();
				} catch (ConnectionLostException e) {
					throw e;
				}
			}

			private void setText1(final String str) {
				// TODO Auto-generated method stub
				runOnUiThread(new Runnable() {
					public void run() {
						mwfb1.setText(str);
					}
				});
			}
			}
		@Override
		protected IOIOLooper createIOIOLooper() {
			return new Looper();

		}
		@Override
		public void onClick(View v) {
			// TODO Auto-generated method stub
			switch (v.getId()) {
			case R.id.mbopen:
				doorpos = true;
				break;
			case R.id.mbclose:
				doorpos = false;
				break;
			default:
				break;
			}
		}
}
```

Automatic menu .xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/particlespic"
    android:orientation="vertical" >

    <Button
        android:id="@+id/bsavemenu"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/bsave"
        android:textSize="@dimen/manualscreentext" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/selectrecipehint"
        android:textSize="@dimen/manualscreentext" />

    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="fill_parent"
        android:textSize="@dimen/manualscreentext"
        android:layout_height="wrap_content" />

    <ExpandableListView
        android:id="@+id/expandableListView1"
        android:layout_width="match_parent"
        android:layout_height="71dp" >

    </ExpandableListView>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <Button
            android:id="@+id/bload"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/bload"
            android:textSize="@dimen/manualscreentext" />

    </LinearLayout>

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:textSize="@dimen/manualscreentext" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="TextView"
        android:textSize="@dimen/manualscreentext" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:textSize="@dimen/manualscreentext" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:textSize="@dimen/manualscreentext" />

</LinearLayout>
```

Automatic menu .java

```
package com.mixer.control;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;

public class AutoOne extends Activity {
	Spinner spinner;
	TextView showme1, showme2, showme3, showme4;

	@Override
	protected void onCreate(Bundle savedInstanceState) {
		// TODO Auto-generated method stub
		super.onCreate(savedInstanceState);
		setContentView(R.layout.automenu);
		spinner = (Spinner) findViewById(R.id.spinner1);
		getRecipenames();
		showme1 = (TextView) findViewById(R.id.textView2);
		showme2 = (TextView) findViewById(R.id.textView3);
```

```
                showme3 = (TextView) findViewById(R.id.textView4);
                showme4 = (TextView) findViewById(R.id.textView5);
                Button savemenu = (Button) findViewById(R.id.bsavemenu);
savemenu.setOnClickListener(new View.OnClickListener() {

                        @Override
                        public void onClick(View v) {
                                // TODO Auto-generated method stub
                                startActivity(new Intent("com.mixer.control.SAVEMENU"));
                        }
                });
Button load = (Button) findViewById(R.id.bload);
load.setOnClickListener(new View.OnClickListener() {

                        @Override
                        public void onClick(View v) {
                                // TODO Auto-generated method stub
                                String selectRecipe = String.valueOf(spinner.getSelectedItem());
                                openRecipe(selectRecipe);
                                Intent i = new Intent("com.mixer.control.AUTORUN");
                                i.putExtra("rez", selectRecipe);
                                startActivity(i);
                                //startActivity(new Intent("com.mixer.control.AUTORUN"));
                        }

                        private void openRecipe(String selectRecipe) {
                                // TODO Auto-generated method stub
                                String valbin1 = "";
                                String valbin2 = "";
                                String valbin3 = "";
                                String valmixtime = "";
                                FileInputStream fis;

                                try {
                                        fis = openFileInput(selectRecipe);
                                        byte[] input = new byte[fis.available()];
                                        while(fis.read(input) != -1){
                                                valbin1 += new String(input);
                                                fis.close();
                                        }
                                }
                                catch (FileNotFoundException e) {
                                        // TODO Auto-generated catch block
                                        e.printStackTrace();
                                } catch (IOException e) {
```

```
        // TODO Auto-generated catch block
        e.printStackTrace();
}
try {
        fis = openFileInput(selectRecipe + "B2");
        byte[] input = new byte[fis.available()];
        while(fis.read(input) != -1){
                valbin2 += new String(input);
                fis.close();
        }
}
catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
} catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
try {
        fis = openFileInput(selectRecipe + "B3");
        byte[] input = new byte[fis.available()];
        while(fis.read(input) != -1){
                valbin3 += new String(input);
                fis.close();
        }
}
catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
} catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
try {
        fis = openFileInput(selectRecipe + "MT");
        byte[] input = new byte[fis.available()];
        while(fis.read(input) != -1){
                valmixtime += new String(input);
                fis.close();
        }
}
catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
} catch (IOException e) {
```

```
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
                showme1.setText(valbin1);
                showme2.setText(valbin2);
                showme3.setText(valbin3);
                showme4.setText(valmixtime);
            }
        });
    }

    private void getRecipenames() {
        // TODO Auto-generated method stub
        String[] recipenames = getApplicationContext().fileList();
        List<String> list = new ArrayList<String>();
        for(int i = 0; i<recipenames.length; i++){
            list.add(recipenames[i]);
        }
        ArrayAdapter<String> recipenameAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_dropdown_item_1line, list);
        spinner.setAdapter(recipenameAdapter);
    }

    @Override
    protected void onPause() {
        // TODO Auto-generated method stub
        super.onPause();
    }
}
```

Save new recipe menu .xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/particlespic"
    android:orientation="vertical" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/resnamehint"
            android:textSize="@dimen/manualscreentext" />

        <EditText
```

```
        android:id="@+id/etrecipename"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:ems="10"
        android:hint="@string/resname"
        android:textSize="@dimen/manualscreentext" >

        <requestFocus />
    </EditText>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/tbman1"
        android:textSize="@dimen/manualscreentext" />

    <EditText
        android:id="@+id/etbin1save"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:ems="10"
        android:hint="@string/bamount1"
        android:textSize="@dimen/manualscreentext" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/tbman2"
        android:textSize="@dimen/manualscreentext" />

    <EditText
        android:id="@+id/etbin2save"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:ems="10"
        android:hint="@string/bamount2"
        android:textSize="@dimen/manualscreentext" >

        <requestFocus />
    </EditText>

</LinearLayout>
```

```
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textView4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/tbman3"
            android:textSize="@dimen/manualscreentext" />

        <EditText
            android:id="@+id/etbin3save"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:hint="@string/bamount3"
            android:textSize="@dimen/manualscreentext" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textView7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/mixtimehint"
            android:textSize="@dimen/manualscreentext" />

        <EditText
            android:id="@+id/etmixtimesave"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:hint="@string/mixtime"
            android:textSize="@dimen/manualscreentext" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <Button
            android:id="@+id/bsave"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="@string/bfinsave"
            android:textSize="@dimen/manualscreentext" />

    </LinearLayout>

</LinearLayout>
```

Save new recipe menu .java

```
package com.mixer.control;

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class SaveMenu extends Activity implements OnClickListener{

	Button bsave;
	EditText Recipename, Bin1, Bin2, Bin3, Mixtime;
	String RECIPENAME, BIN1, BIN2, BIN3, MIXTIME;

	@Override
	protected void onCreate(Bundle savedInstanceState) {
		// TODO Auto-generated method stub
		super.onCreate(savedInstanceState);
		setContentView(R.layout.autosave);
		bsave = (Button) findViewById(R.id.bsave);
		bsave.setOnClickListener(this);
		Recipename = (EditText) findViewById(R.id.etrecipename);
		Bin1 = (EditText) findViewById(R.id.etbin1save);
		Bin2 = (EditText) findViewById(R.id.etbin2save);
		Bin3 = (EditText) findViewById(R.id.etbin3save);
		Mixtime = (EditText) findViewById(R.id.etmixtimesave);
	}

	@Override
	protected void onPause() {
		// TODO Auto-generated method stub
		super.onPause();
	}

	@Override
	public void onClick(View v) {
		// TODO Auto-generated method stub
		RECIPENAME = Recipename.getText().toString();
		if (RECIPENAME.contentEquals("")) {
```

```
				RECIPENAME = "UNTITLED";
			}
			BIN1 = Bin1.getText().toString();
			BIN2 = Bin2.getText().toString();
			BIN3 = Bin3.getText().toString();
			MIXTIME = Mixtime.getText().toString();

			try {
				FileOutputStream fos = openFileOutput(RECIPENAME, MODE_PRIVATE);
				fos.write(BIN1.getBytes());
				fos.close();
			} catch (FileNotFoundException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			} catch (IOException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
			try {
				FileOutputStream fos = openFileOutput(RECIPENAME + "B2",
MODE_PRIVATE);
				fos.write(BIN2.getBytes());
				fos.close();
			} catch (FileNotFoundException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			} catch (IOException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
			try {
				FileOutputStream fos = openFileOutput(RECIPENAME + "B3",
MODE_PRIVATE);
				fos.write(BIN3.getBytes());
				fos.close();
			} catch (FileNotFoundException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			} catch (IOException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
			try {
				FileOutputStream fos = openFileOutput(RECIPENAME + "MT",
MODE_PRIVATE);
```

```
                    fos.write(MIXTIME.getBytes());
                    fos.close();
            } catch (FileNotFoundException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            }

            startActivity(new Intent("com.mixer.control.AUTOONE"));
        }
}
```

Running a preset recipe .xml (Fifth level of control program)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/particlespic"
    android:orientation="vertical" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/Namehint"
            android:textSize="@dimen/manualscreentext" />

        <TextView
            android:id="@+id/atvrecipename"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/resname"
            android:textSize="@dimen/manualscreentext" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textView4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/tbman1"
            android:textSize="@dimen/manualscreentext" />

        <TextView
            android:id="@+id/atvb1"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/bamount1"
            android:textSize="@dimen/manualscreentext" />

        <EditText
            android:id="@+id/etabin1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:hint="@string/cw1"
            android:textSize="@dimen/manualscreentext" >

            <requestFocus />
        </EditText>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textView6"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/tbman2"
            android:textSize="@dimen/manualscreentext" />

        <TextView
            android:id="@+id/atvb2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/bamount2"
            android:textSize="@dimen/manualscreentext" />

        <EditText
            android:id="@+id/editText2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:hint="@string/cw2"
            android:textSize="@dimen/manualscreentext" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textView8"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/tbman3"
            android:textSize="@dimen/manualscreentext" />
```

```xml
    <TextView
        android:id="@+id/atvb3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/bamount3"
        android:textSize="@dimen/manualscreentext" />

    <EditText
        android:id="@+id/editText3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:ems="10"
        android:hint="@string/cw3"
        android:textSize="@dimen/manualscreentext" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/textView13"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/mixtimehint"
        android:textSize="@dimen/manualscreentext" />

    <EditText
        android:id="@+id/aetmt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:ems="10"
        android:hint="@string/mixtime"
        android:text="@string/mixtime"
        android:textSize="@dimen/manualscreentext" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/textView14"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/tchainman"
        android:textSize="@dimen/manualscreentext" />

    <EditText
        android:id="@+id/aetchainconvstatus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:ems="10"
        android:hint="@string/chainconvstatus"
```

```xml
            android:textSize="@dimen/manualscreentext" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textView15"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/manmixdoor"
            android:textSize="@dimen/manualscreentext" />

        <EditText
            android:id="@+id/aetmixdoorstatus"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:hint="@string/mixdoorstatus"
            android:textSize="@dimen/manualscreentext" />

        <TextView
            android:id="@+id/textView3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/manmix"
            android:textSize="@dimen/manualscreentext" />

        <EditText
            android:id="@+id/aetmixstatus"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:hint="@string/mixstatus"
            android:textSize="@dimen/manualscreentext" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <Button
            android:id="@+id/bastart"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Start"
            android:textSize="@dimen/manualscreentext" />

        <Button
            android:id="@+id/bastop"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
```

```
            android:text="Stop"
            android:textSize="@dimen/manualscreentext" />

    </LinearLayout>

</LinearLayout>
```

Running a preset recipe .java (Fifth level of control program)

```
package com.mixer.control;

import ioio.lib.api.AnalogInput;
import ioio.lib.api.DigitalOutput;
import ioio.lib.api.PwmOutput;
import ioio.lib.api.exception.ConnectionLostException;
import ioio.lib.util.BaseIOIOLooper;
import ioio.lib.util.IOIOLooper;
import ioio.lib.util.android.IOIOActivity;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class AutoRun extends IOIOActivity implements OnClickListener{

	TextView arecipename, atvbin1, atvbin2, atvbin3;
	EditText aetmixtime, aetchainstat, aetmdoorstat, aetmixstat, etautobin1;
	Button bastart, bastop;
	Boolean bin1, chain, mixdoor, mix = false;
	Boolean startsystem, start = true;
	float finalw1, finalw2, finalw3;
	String weigh = "a";
	String lockweight = "b";

	@Override
	protected void onCreate(Bundle savedInstanceState) {
		// TODO Auto-generated method stub
		super.onCreate(savedInstanceState);
		setContentView(R.layout.autorun);
		arecipename = (TextView) findViewById(R.id.atvrecipename);
```

```
            atvbin1 = (TextView) findViewById(R.id.atvb1);
            atvbin2 = (TextView) findViewById(R.id.atvb2);
            atvbin3 = (TextView) findViewById(R.id.atvb3);
            aetmixtime = (EditText) findViewById(R.id.aetmt);
            aetchainstat = (EditText) findViewById(R.id.aetchainconvstatus);
            aetmdoorstat = (EditText) findViewById(R.id.aetmixdoorstatus);
            aetmixstat = (EditText) findViewById(R.id.aetmixstatus);
            etautobin1 = (EditText) findViewById(R.id.etabin1);
            bastart = (Button) findViewById(R.id.bastart);
            bastart.setOnClickListener(this);
            bastop = (Button) findViewById(R.id.bastop);
            bastop.setOnClickListener(this);
            aetmdoorstat.setText("Door is open");
            Intent i = getIntent();
            String msg = i.getStringExtra("rez");
            loadr(msg);
        }
//Load chosen recipe settings into new class
        private void loadr(String msg) {
            // TODO Auto-generated method stub
            String valbin1 = "";
            String valbin2 = "";
            String valbin3 = "";
            String valmixtime = "";
            FileInputStream fis;

            try {
                fis = openFileInput(msg);
                byte[] input = new byte[fis.available()];
                while(fis.read(input) != -1){
                    valbin1 += new String(input);
                    fis.close();
                }
            }
            catch (FileNotFoundException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            try {
                fis = openFileInput(msg + "B2");
                byte[] input = new byte[fis.available()];
                while(fis.read(input) != -1){
```

```
				valbin2 += new String(input);
				fis.close();
			}
		}
		catch (FileNotFoundException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		try {
			fis = openFileInput(msg + "B3");
			byte[] input = new byte[fis.available()];
			while(fis.read(input) != -1){
				valbin3 += new String(input);
				fis.close();
			}
		}
		catch (FileNotFoundException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		try {
			fis = openFileInput(msg + "MT");
			byte[] input = new byte[fis.available()];
			while(fis.read(input) != -1){
				valmixtime += new String(input);
				fis.close();
			}
		}
		catch (FileNotFoundException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		atvbin1.setText(valbin1);
		atvbin2.setText(valbin2);
		atvbin3.setText(valbin3);
		aetmixtime.setText(valmixtime);
```

```
    }

    //Start the looper to control automatic mixing
    class Looper extends BaseIOIOLooper{
        private DigitalOutput FB1_PIN;
        private DigitalOutput CHAINCONV_PIN;
        private DigitalOutput MIX_PIN;
        private AnalogInput FB1W_PIN;
        private PwmOutput MIXDOOR_PIN;

        @Override
        protected void setup() throws ConnectionLostException, InterruptedException {
            FB1_PIN = ioio_.openDigitalOutput(34, false);
            CHAINCONV_PIN = ioio_.openDigitalOutput(35, true);
            MIX_PIN = ioio_.openDigitalOutput(36, false);
            FB1W_PIN = ioio_.openAnalogInput(40);
            MIXDOOR_PIN = ioio_.openPwmOutput(3, 100);
        }

        @Override
        public void loop() throws ConnectionLostException, InterruptedException {
            try {
                Thread.sleep(300);
            } catch (InterruptedException e) {
            }
            try {
                final float reading1 = FB1W_PIN.read();
                setText1(Float.toString((reading1*100)));

            } catch (InterruptedException e) {
                ioio_.disconnect();
            } catch (ConnectionLostException e) {
                throw e;
            }

            if(start == true) {
                Thread.sleep(10);
                if("a".equals(weigh)){
                    if("b".equals(lockweight)){
                        final float reading1 = FB1W_PIN.read();
                        setText1(Float.toString((reading1*100)));
                        finalw1 = (reading1 * 100) - Integer.valueOf((String)
atvbin1.getText());

                        lockweight = "Z";
                        MIX_PIN.write(false);
```

```
                    final String nomixrun = "Mixer not in operation";
                    setText3(nomixrun);
                    MIXDOOR_PIN.setPulseWidth(920);
                    final String dooropen = "Door is open";
                    setText2(dooropen);
                    CHAINCONV_PIN.write(true);
                    final String chainrun = "Chain conveyor is in
operation";

                    setText4(chainrun);
                } else {
                final float reading1 = FB1W_PIN.read();
                setText1(Float.toString((reading1*100)));
                if((reading1 *100)>finalw1){
                    FB1_PIN.write(true);
                }else {
                    FB1_PIN.write(false);
                    Thread.sleep(3000);
                    CHAINCONV_PIN.write(false);
                    final String chainrun = "Chain conveyor is not in
operation";

                    setText4(chainrun);
                    MIXDOOR_PIN.setPulseWidth(1400);
                    final String doorclose = "Door is closed";
                    setText2(doorclose);
                    Thread.sleep(1500);
                    lockweight = "b";
                    weigh = "X";

                }
                }}
            else {
                CHAINCONV_PIN.write(false);
                final String chainrun = "Chain conveyor is not in operation";
                setText4(chainrun);
                MIXDOOR_PIN.setPulseWidth(1400);
                final String doorclose = "Door is closed";
                setText2(doorclose);
                MIX_PIN.write(true);
                final String mixrun = "Mixer in operation";
                setText3(mixrun);
                String recmixtime = aetmixtime.getText().toString();
                Thread.sleep((long) Float.parseFloat(recmixtime));
                MIX_PIN.write(false);
                final String nomixrun = "Mixer not in operation";
                setText3(nomixrun);
```

```
                        Thread.sleep(1000);
                        MIXDOOR_PIN.setPulseWidth(920);
                        final String dooropen = "Door is open";
                        setText2(dooropen);
                        Thread.sleep(1000);
                        MIXDOOR_PIN.setPulseWidth(1400);
                        setText2(doorclose);
                        Thread.sleep(1000);
                        MIX_PIN.write(true);
                        setText3(mixrun);
                        Thread.sleep(1000);
                        MIX_PIN.write(false);
                        setText3(nomixrun);
                        MIXDOOR_PIN.setPulseWidth(920);
                        setText2(dooropen);
                        Thread.sleep(1000);
                        weigh = "a";
                }

        } else {
                MIXDOOR_PIN.setPulseWidth(1400);
                final String doorclose = "Door is closed";
                setText2(doorclose);
                CHAINCONV_PIN.write(false);
                final String chainrun = "Chain conveyor is not in operation";
                setText4(chainrun);
                FB1_PIN.write(false);
                MIX_PIN.write(false);
                final String nomixrun = "Mixer not in operation";
                setText3(nomixrun);
                weigh = "a";
                lockweight = "b";
                Thread.sleep(300);
        }
}

private void setText4(final String str) {
        // TODO Auto-generated method stub
        runOnUiThread(new Runnable() {
                public void run() {
                        aetchainstat.setText(str);
                }
        });
}
```

```
		private void setText3(final String str) {
			// TODO Auto-generated method stub
			runOnUiThread(new Runnable() {
				public void run() {
					aetmixstat.setText(str);
				}
			});
		}

		private void setText2(final String str) {
			// TODO Auto-generated method stub
			runOnUiThread(new Runnable() {
				public void run() {
					aetmdoorstat.setText(str);
				}
			});
		}

		private void setText1(final String str) {
			// TODO Auto-generated method stub
			runOnUiThread(new Runnable() {
				public void run() {
					etautobin1.setText(str);
				}
			});
		}
		}
	@Override
	protected IOIOLooper createIOIOLooper() {
		return new Looper();
	}

	@Override
	protected void onPause() {
		// TODO Auto-generated method stub
		super.onPause();
	}

	@Override
	public void onClick(View v) {
		// TODO Auto-generated method stub
		switch (v.getId()) {
		case R.id.bastart:
			start = true;
			break;
```

```
			case R.id.bastop:
				start = false;
				break;
			default:
				break;
		}
		}
}
```

Appendix C – Detail drawings of the V- and Y-blenders and supporting frame

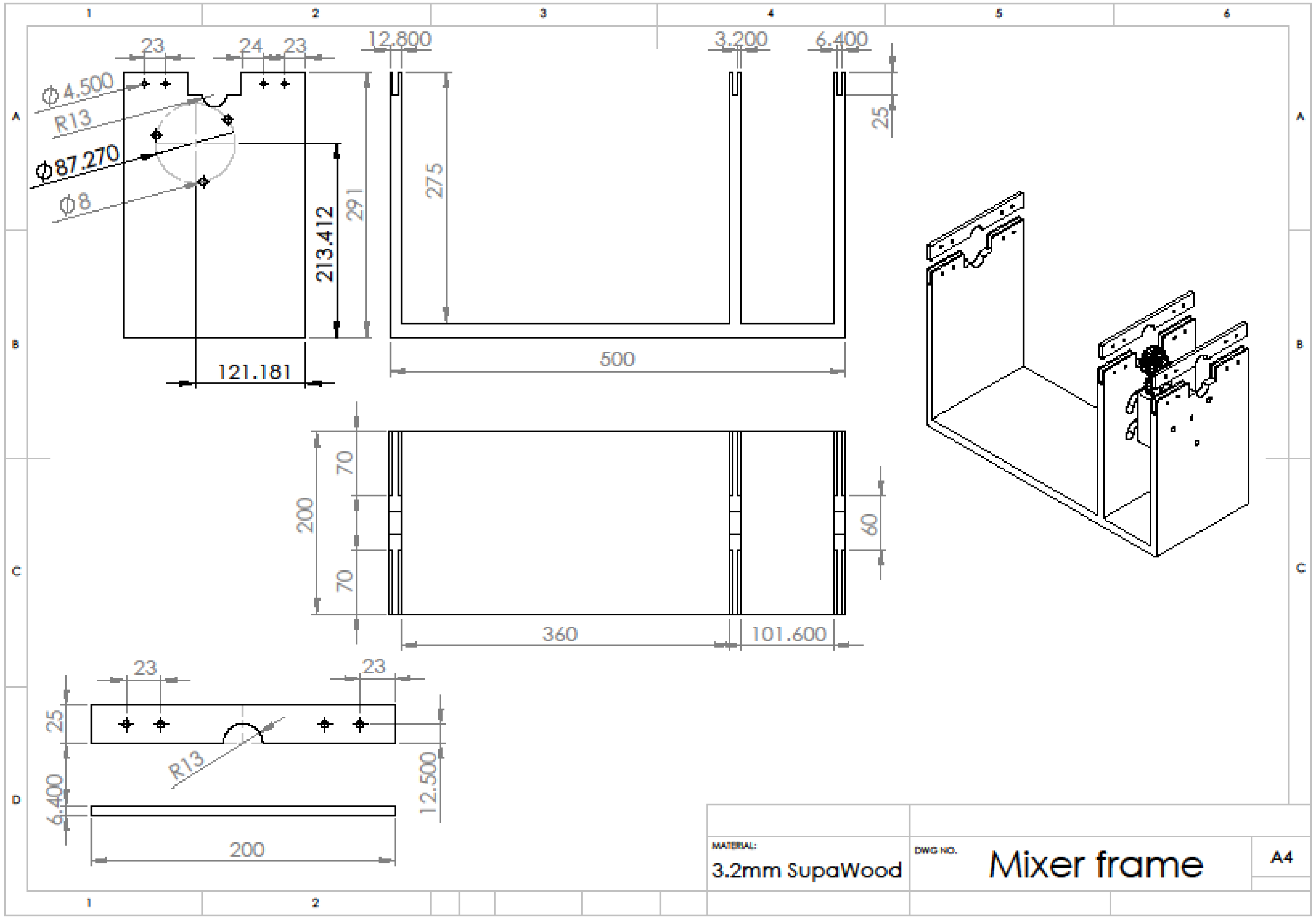

Figure 52: Detail drawing of mixer frame.

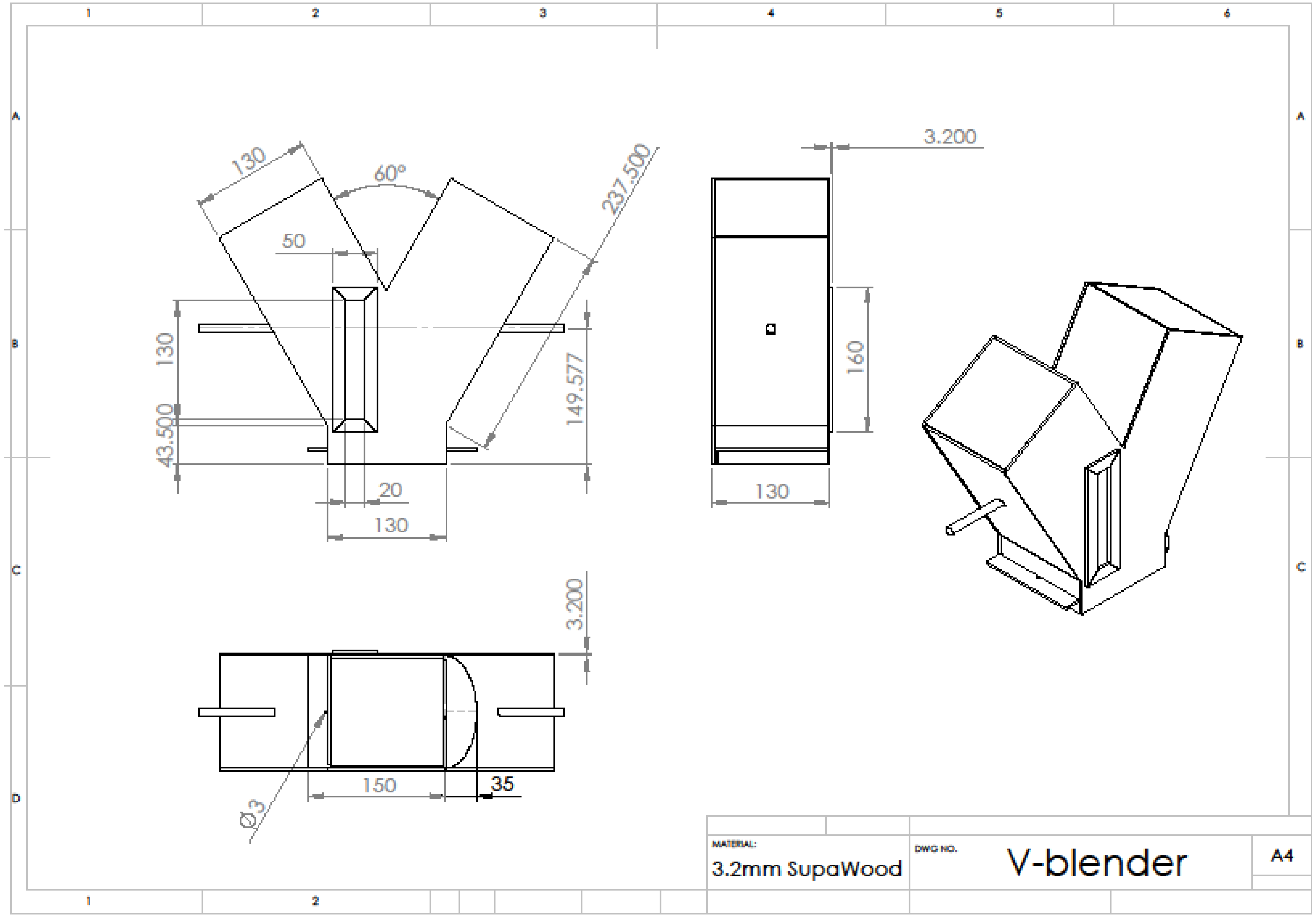

Figure 53: Detail drawing of V-blender,

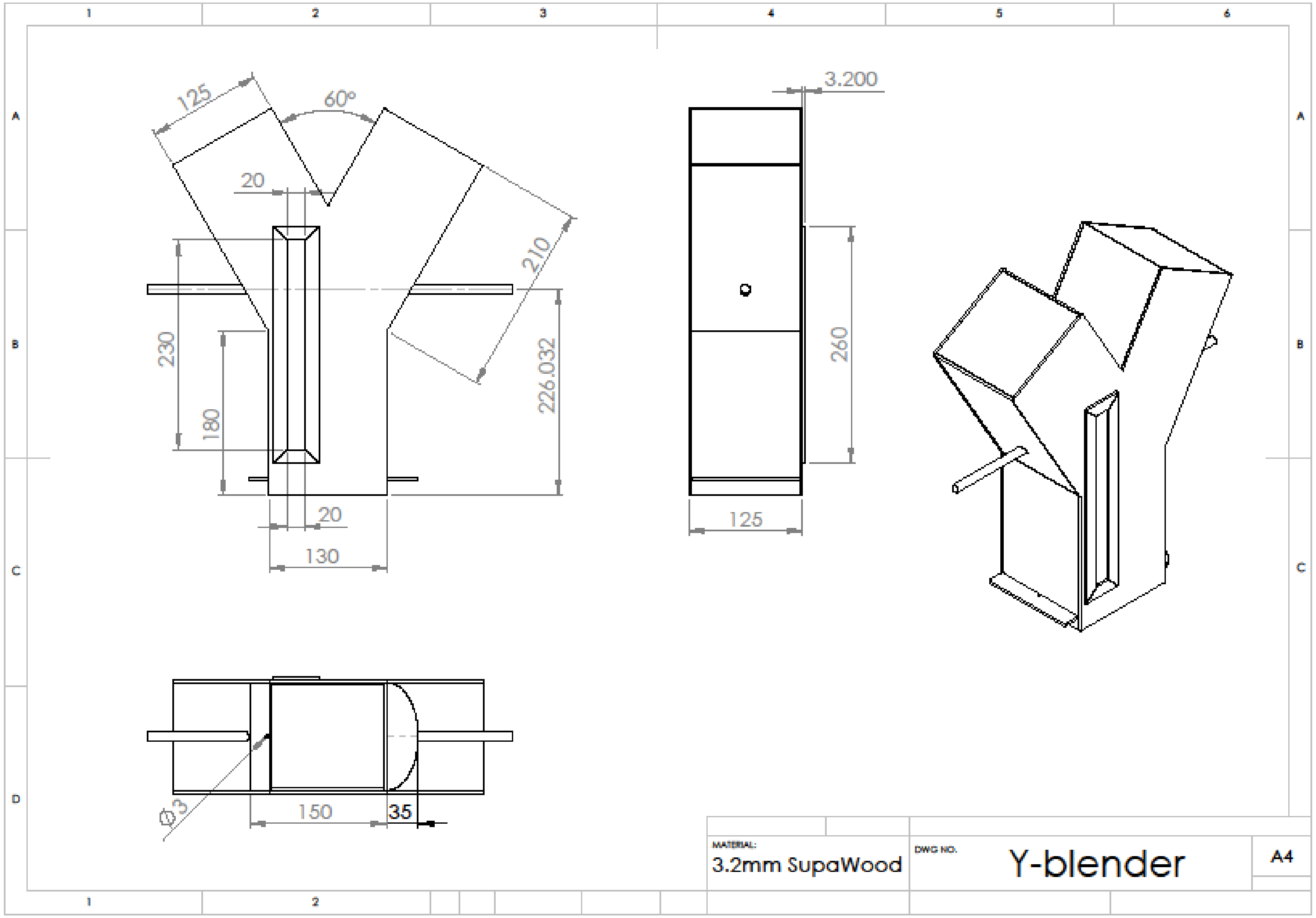

Figure 54: Detail drawing of Y-blender.

Appendix D – Data acquired during filtration evaluation method

Table 9: Data acquired during the filtration method for the Y-blender filled to 30%.

Y-Blender @ 30% fill																
Sample number #	Filter weight (g)	Filter & sample (g)	After H2O (g)	Sample weight (g)	Combination (%)	Maize (%)	Maize remaining (g)	Combination remaining (g)	Total remaining (g)	Maize remaining (%)	Standard deviation (%)	Standard deviation (ratio)	Standard deviation of total mixture	Acceptable deviation limit	Mixing index	Revolution number
2.1	1.27	4.03	3.39	2.76	23.188	76.812	939.600	140.400	1080.000	87.000	5.9409	0.0594	0.3363	0.0020	0.1717	2
2.2	1.26	6.46	5.73	5.20	14.038	85.962	937.480	139.760	1077.240	87.026						
2.3	1.28	3.06	2.58	1.78	26.966	73.034	933.010	139.030	1072.040	87.031						
2.4	1.26	4.92	3.96	3.66	26.230	73.770	931.710	138.550	1070.260	87.055						
4.1	1.27	4.04	3.53	2.77	18.412	81.588	929.010	137.590	1066.600	87.100	3.6754	0.0368	0.3352	0.0020	0.1043	4
4.2	1.29	2.67	2.51	1.38	11.594	88.406	926.750	137.080	1063.830	87.114						
4.3	1.26	3.90	3.60	2.64	11.364	88.636	925.530	136.920	1062.450	87.113						
4.4	1.29	4.07	3.78	2.78	10.432	89.568	923.190	136.620	1059.810	87.109						
6.1	1.28	3.38	3.10	2.10	13.333	86.667	920.700	136.330	1057.030	87.103	2.1922	0.0219	0.3352	0.0020	0.0598	6
6.2	1.24	3.68	3.36	2.44	13.115	86.885	918.880	136.050	1054.930	87.103						
6.3	1.28	2.47	2.26	1.19	17.647	82.353	916.760	135.730	1052.490	87.104						
6.4	1.29	4.21	3.82	2.92	13.356	86.644	915.780	135.520	1051.300	87.109						
8.1	1.26	3.53	3.20	2.27	14.537	85.463	913.250	135.130	1048.380	87.111	2.2772	0.0228	0.3351	0.0020	0.0624	8
8.2	1.27	4.00	3.59	2.73	15.018	84.982	911.310	134.800	1046.110	87.114						
8.3	1.29	2.94	2.77	1.65	10.303	89.697	908.990	134.390	1043.380	87.120						
8.4	1.24	3.22	2.99	1.98	11.616	88.384	907.510	134.220	1041.730	87.116						
10.1	1.28	3.00	2.78	1.72	12.791	87.209	905.760	133.990	1039.750	87.113	1.4595	0.0146	0.3351	0.0020	0.0378	10
10.2	1.28	3.91	3.53	2.63	14.449	85.551	904.260	133.770	1038.030	87.113						
10.3	1.25	4.37	4.01	3.12	11.538	88.462	902.010	133.390	1035.400	87.117						
10.4	1.26	5.53	5.05	4.27	11.241	88.759	899.250	133.030	1032.280	87.113						
15.1	1.31	5.01	4.49	3.70	14.054	85.946	895.460	132.550	1028.010	87.106	0.7378	0.0074	0.3351	0.0020	0.0161	15
15.2	1.27	3.59	3.26	2.32	14.224	85.776	892.280	132.030	1024.310	87.110						
15.3	1.28	5.81	5.21	4.53	13.245	86.755	890.290	131.700	1021.990	87.113						
15.4	1.30	3.56	3.22	2.26	15.044	84.956	886.360	131.100	1017.460	87.115						
20.1	1.26	4.28	3.84	3.02	14.570	85.430	884.440	130.760	1015.200	87.120	1.7375	0.0174	0.3350	0.0020	0.0462	20
20.2	1.24	3.49	3.20	2.25	12.889	87.111	881.860	130.320	1012.180	87.125						
20.3	1.31	3.01	2.82	1.70	11.176	88.824	879.900	130.030	1009.930	87.125						
20.4	1.28	3.97	3.68	2.69	10.781	89.219	878.390	129.840	1008.230	87.122						
25.1	1.27	3.37	3.12	2.10	11.905	88.095	875.990	129.550	1005.540	87.116	0.7806	0.0078	0.3350	0.0020	0.0174	25
25.2	1.30	4.81	4.33	3.51	13.675	86.325	874.140	129.300	1003.440	87.114						
25.3	1.28	4.09	3.72	2.81	13.167	86.833	871.110	128.820	999.930	87.117						
25.4	1.26	3.80	3.46	2.54	13.386	86.614	868.670	128.450	997.120	87.118						

Table 10: Data acquired during the filtration method for the Y-blender filled to 30% continued.

Y-Blender @ 30% fill																
Sample number #	Filter weight (g)	Filter & sample (g)	After H2O (g)	Sample weight (g)	Combination (%)	Maize (%)	Maize remaining (g)	Combination remaining (g)	Total remaining (g)	Maize remaining (%)	Standard deviation (%)	Standard deviation (ratio)	Standard deviation of total mixture	Acceptable deviation limit	Mixing index	Revolution number
30.1	1.30	4.61	4.13	3.31	14.502	85.498	866.470	128.110	994.580	87.119	0.6150	0.0062	0.3350	0.0020	0.0125	30
30.2	1.29	4.70	4.23	3.41	13.783	86.217	863.640	127.630	991.270	87.125						
30.3	1.30	4.29	3.85	2.99	14.716	85.284	860.700	127.160	987.860	87.128						
30.4	1.29	5.47	4.91	4.18	13.397	86.603	858.150	126.720	984.870	87.133						
35.1	1.29	4.74	4.38	3.45	10.435	89.565	854.530	126.160	980.690	87.136	0.9580	0.0096	0.3348	0.0020	0.0228	35
35.2	1.30	5.37	4.90	4.07	11.548	88.452	851.440	125.800	977.240	87.127						
35.3	1.31	4.22	3.87	2.91	12.027	87.973	847.840	125.330	973.170	87.121						
35.4	1.27	3.63	3.33	2.36	12.712	87.288	845.280	124.980	970.260	87.119						
40.1	1.28	4.28	3.92	3.00	12.000	88.000	843.220	124.680	967.900	87.119	0.5337	0.0053	0.3350	0.0020	0.0100	40
40.2	1.28	4.61	4.18	3.33	12.913	87.087	840.580	124.320	964.900	87.116						
40.3	1.30	3.51	3.24	2.21	12.217	87.783	837.680	123.890	961.570	87.116						
40.4	1.30	3.36	3.09	2.06	13.107	86.893	835.740	123.620	959.360	87.114						
50.1	1.31	5.26	4.80	3.95	11.646	88.354	833.950	123.350	957.300	87.115	0.6187	0.0062	0.3350	0.0020	0.0126	50
50.2	1.26	4.91	4.43	3.65	13.151	86.849	830.460	122.890	953.350	87.110						
50.3	1.28	4.12	3.77	2.84	12.324	87.676	827.290	122.410	949.700	87.111						
50.4	1.31	3.63	3.34	2.32	12.500	87.500	824.800	122.060	946.860	87.109						
60.1	1.27	4.55	4.11	3.28	13.415	86.585	822.770	121.770	944.540	87.108	0.9197	0.0092	0.3351	0.0020	0.0216	60
60.2	1.26	5.36	4.84	4.10	12.683	87.317	819.930	121.330	941.260	87.110						
60.3	1.30	5.01	4.51	3.71	13.477	86.523	816.350	120.810	937.160	87.109						
60.4	1.26	3.41	3.09	2.15	14.884	85.116	813.140	120.310	933.450	87.111						
80.1	1.28	5.58	5.00	4.30	13.488	86.512	811.310	119.990	931.300	87.116	1.5876	0.0159	0.3350	0.0020	0.0417	80
80.2	1.30	5.41	4.85	4.11	13.625	86.375	807.590	119.410	927.000	87.119						
80.3	1.28	3.01	2.79	1.73	12.717	87.283	804.040	118.850	922.890	87.122						
80.4	1.27	5.68	5.23	4.41	10.204	89.796	802.530	118.630	921.160	87.122						
100.1	1.27	3.60	3.25	2.33	15.021	84.979	798.570	118.180	916.750	87.109	0.8667	0.0087	0.3351	0.0020	0.0200	100
100.2	1.28	5.23	4.66	3.95	14.430	85.570	796.590	117.830	914.420	87.114						
100.3	1.26	3.69	3.29	2.43	16.461	83.539	793.210	117.260	910.470	87.121						
100.4	1.30	4.31	3.84	3.01	15.615	84.385	791.180	116.860	908.040	87.131						

Table 11: Data acquired during the filtration method for the Y-blender filled to 60%.

Y-Blender @ 60% fill																
Sample number #	Filter weight (g)	Filter & sample (g)	After H2O (g)	Sample weight (g)	Combination (%)	Maize (%)	Maize remaining (g)	Combination remaining (g)	Total remaining (g)	Maize remaining (%)	Standard deviation (%)	Standard deviation (ratio)	Standard deviation of total mixture	Acceptable deviation limit	Mixing index	Revolution number
2.1	1.26	3.90	3.37	2.64	20.076	79.924	1879.200	280.800	2160.000	87.000	26.4381	0.2644	0.3363	0.0020	0.7849	2
2.2	1.24	5.72	3.15	4.48	57.366	42.634	1877.090	280.270	2157.360	87.009						
2.3	1.28	4.60	4.56	3.32	1.205	98.795	1875.180	277.700	2152.880	87.101						
2.4	1.26	4.20	4.16	2.94	1.361	98.639	1871.900	277.660	2149.560	87.083						
4.1	1.26	3.30	3.16	2.04	6.863	93.137	1869.000	277.620	2146.620	87.067	1.7548	0.0175	0.3356	0.0020	0.0466	4
4.2	1.28	5.26	5.00	3.98	6.533	93.467	1867.100	277.480	2144.580	87.061						
4.3	1.26	4.20	3.97	2.94	7.823	92.177	1863.380	277.220	2140.600	87.049						
4.4	1.26	5.54	5.38	4.28	3.738	96.262	1860.670	276.990	2137.660	87.042						
6.1	1.29	3.93	3.49	2.64	16.667	83.333	1856.550	276.830	2133.380	87.024	1.4812	0.0148	0.3360	0.0020	0.0384	6
6.2	1.25	4.36	3.94	3.11	13.505	86.495	1854.350	276.390	2130.740	87.028						
6.3	1.25	3.52	3.17	2.27	15.419	84.581	1851.660	275.970	2127.630	87.029						
6.4	1.25	4.50	3.96	3.25	16.615	83.385	1849.740	275.620	2125.360	87.032						
8.1	1.25	3.51	3.25	2.26	11.504	88.496	1847.030	275.080	2122.110	87.037	2.3969	0.0240	0.3359	0.0020	0.0658	8
8.2	1.26	5.18	4.65	3.92	13.520	86.480	1845.030	274.820	2119.850	87.036						
8.3	1.26	5.30	4.91	4.04	9.653	90.347	1841.640	274.290	2115.930	87.037						
8.4	1.26	3.37	3.05	2.11	15.166	84.834	1837.990	273.900	2111.890	87.031						
10.1	1.27	3.45	3.14	2.18	14.220	85.780	1836.200	273.580	2109.780	87.033	2.0662	0.0207	0.3359	0.0020	0.0559	10
10.2	1.26	6.59	6.04	5.33	10.319	89.681	1834.330	273.270	2107.600	87.034						
10.3	1.26	3.60	3.25	2.34	14.957	85.043	1829.550	272.720	2102.270	87.027						
10.4	1.28	4.09	3.70	2.81	13.879	86.121	1827.560	272.370	2099.930	87.030						
15.1	1.29	4.49	4.01	3.20	15.000	85.000	1825.140	271.980	2097.120	87.031	0.4828	0.0048	0.3360	0.0020	0.0085	15
15.2	1.24	3.87	3.48	2.63	14.829	85.171	1822.420	271.500	2093.920	87.034						
15.3	1.26	4.33	3.86	3.07	15.309	84.691	1820.180	271.110	2091.290	87.036						
15.4	1.29	3.99	3.56	2.70	15.926	84.074	1817.580	270.640	2088.220	87.040						
20.1	1.28	4.50	4.00	3.22	15.528	84.472	1815.310	270.210	2085.520	87.044	0.9633	0.0096	0.3358	0.0020	0.0229	20
20.2	1.28	5.14	4.62	3.86	13.472	86.528	1812.590	269.710	2082.300	87.047						
20.3	1.29	4.13	3.69	2.84	15.493	84.507	1809.250	269.190	2078.440	87.048						
20.4	1.24	6.00	5.30	4.76	14.706	85.294	1806.850	268.750	2075.600	87.052						
25.1	1.28	3.87	3.44	2.59	16.602	83.398	1802.790	268.050	2070.840	87.056	1.6971	0.0170	0.3357	0.0020	0.0449	25
25.2	1.24	4.97	4.42	3.73	14.745	85.255	1800.630	267.620	2068.250	87.061						
25.3	1.26	4.74	4.23	3.48	14.655	85.345	1797.450	267.070	2064.520	87.064						
25.4	1.26	6.64	5.97	5.38	12.454	87.546	1794.480	266.560	2061.040	87.067						

Table 12: Data acquired during the filtration method for the Y-blender filled to 60% continued.

Y-Blender @ 60% fill																
Sample number #	Filter weight (g)	Filter & sample (g)	After H2O (g)	Sample weight (g)	Combination (%)	Maize (%)	Maize remaining (g)	Combination remaining (g)	Total remaining (g)	Maize remaining (%)	Standard deviation (%)	Standard deviation (ratio)	Standard deviation of total mixture	Acceptable deviation limit	Mixing index	Revolution number
30.1	1.30	4.36	3.86	3.06	16.340	83.660	1789.770	265.890	2055.660	87.065	0.7038	0.0070	0.3356	0.0020	0.0151	30
30.2	1.27	4.73	4.19	3.46	15.607	84.393	1787.210	265.390	2052.600	87.071						
30.3	1.29	3.68	3.33	2.39	14.644	85.356	1784.290	264.850	2049.140	87.075						
30.4	1.30	4.96	4.40	3.66	15.301	84.699	1782.250	264.500	2046.750	87.077						
35.1	1.27	4.13	3.60	2.86	18.531	81.469	1779.150	263.940	2043.090	87.081	1.8642	0.0186	0.3354	0.0020	0.0499	35
35.2	1.26	4.69	4.12	3.43	16.618	83.382	1776.820	263.410	2040.230	87.089						
35.3	1.30	3.98	3.58	2.68	14.925	85.075	1773.960	262.840	2036.800	87.095						
35.4	1.26	3.62	3.28	2.36	14.407	85.593	1771.680	262.440	2034.120	87.098						
40.1	1.25	4.36	3.85	3.11	16.399	83.601	1769.660	262.100	2031.760	87.100	1.3760	0.0138	0.3352	0.0020	0.0353	40
40.2	1.27	4.69	4.23	3.42	13.450	86.550	1767.060	261.590	2028.650	87.105						
40.3	1.29	5.00	4.40	3.71	16.173	83.827	1764.100	261.130	2025.230	87.106						
40.4	1.25	5.26	4.67	4.01	14.713	85.287	1760.990	260.530	2021.520	87.112						
50.1	1.30	5.26	4.62	3.96	16.162	83.838	1757.570	259.940	2017.510	87.116	2.0451	0.0205	0.3350	0.0020	0.0554	50
50.2	1.27	5.02	4.38	3.75	17.067	82.933	1754.250	259.300	2013.550	87.122						
50.3	1.29	3.57	3.17	2.28	17.544	82.456	1751.140	258.660	2009.800	87.130						
50.4	1.32	6.09	5.47	4.77	12.998	87.002	1749.260	258.260	2007.520	87.135						
60.1	1.27	4.97	4.43	3.70	14.595	85.405	1745.110	257.640	2002.750	87.136	1.2701	0.0127	0.3348	0.0020	0.0322	60
60.2	1.26	4.18	3.71	2.92	16.096	83.904	1741.950	257.100	1999.050	87.139						
60.3	1.29	6.37	5.62	5.08	14.764	85.236	1739.500	256.630	1996.130	87.144						
60.4	1.26	5.42	4.70	4.16	17.308	82.692	1735.170	255.880	1991.050	87.148						
80.1	1.26	4.28	3.76	3.02	17.219	82.781	1731.730	255.160	1986.890	87.158	5.5121	0.0551	0.3346	0.0020	0.1597	80
80.2	1.30	7.63	7.30	6.33	5.213	94.787	1729.230	254.640	1983.870	87.164						
80.3	1.30	5.12	4.54	3.82	15.183	84.817	1723.230	254.310	1977.540	87.140						
80.4	1.28	4.42	3.92	3.14	15.924	84.076	1719.990	253.730	1973.720	87.145						
100.1	1.31	4.98	4.34	3.67	17.439	82.561	1717.350	253.230	1970.580	87.149	2.1672	0.0217	0.3347	0.0020	0.0591	100
100.2	1.28	7.31	6.57	6.03	12.272	87.728	1714.320	252.590	1966.910	87.158						
100.3	1.27	4.20	3.75	2.93	15.358	84.642	1709.030	251.850	1960.880	87.156						
100.4	1.29	5.19	4.57	3.90	15.897	84.103	1706.550	251.400	1957.950	87.160						

Table 13: Data acquired during the filtration method for the V-blender filled to 30%.

V-Blender @ 30% fill																	
Sample number #	Filter weight (g)	Filter & sample (g)	After H2O (g)	Sample weight (g)	Combination (%)	Maize (%)	Maize remaining (g)	Combination remaining (g)	Total remaining (g)	Maize remaining (%)	Standard deviation (%)	Standard deviation (ratio)	Standard deviation of total mixture	Acceptable deviation limit	Mixing index	Revolution number
2.1	1.27	3.43	3.13	2.16	13.889	86.111	853.470	127.530	981.000	87.000	5.4022	0.0540	0.3363	0.0020	0.1556	2
2.2	1.23	5.92	4.72	4.69	25.586	74.414	851.610	127.230	978.840	87.002						
2.3	1.27	3.57	3.22	2.30	15.217	84.783	848.120	126.030	974.150	87.063						
2.4	1.30	3.21	2.81	1.91	20.942	79.058	846.170	125.680	971.850	87.068						
4.1	1.26	4.32	3.84	3.06	15.686	84.314	844.660	125.280	969.940	87.084	5.1551	0.0516	0.3354	0.0020	0.1486	4
4.2	1.26	6.08	5.52	4.82	11.618	88.382	842.080	124.800	966.880	87.093						
4.3	1.28	4.02	3.44	2.74	21.168	78.832	837.820	124.240	962.060	87.086						
4.4	1.31	3.54	3.03	2.23	22.870	77.130	835.660	123.660	959.320	87.110						
6.1	1.24	6.46	5.48	5.22	18.774	81.226	833.940	123.150	957.090	87.133	4.8009	0.0480	0.3348	0.0020	0.1382	6
6.2	1.28	6.09	5.59	4.81	10.395	89.605	829.700	122.170	951.870	87.165						
6.3	1.25	5.11	4.28	3.86	21.503	78.497	825.390	121.670	947.060	87.153						
6.4	1.26	4.22	3.77	2.96	15.203	84.797	822.360	120.840	943.200	87.188						
8.1	1.28	3.76	3.18	2.48	23.387	76.613	819.850	120.390	940.240	87.196	3.7297	0.0373	0.3341	0.0020	0.1063	8
8.2	1.32	4.65	4.12	3.33	15.916	84.084	817.950	119.810	937.760	87.224						
8.3	1.24	3.47	2.97	2.23	22.422	77.578	815.150	119.280	934.430	87.235						
8.4	1.30	3.63	3.23	2.33	17.167	82.833	813.420	118.780	932.200	87.258						
10.1	1.28	4.24	3.64	2.96	20.270	79.730	811.490	118.380	929.870	87.269	3.0012	0.0300	0.3333	0.0020	0.0845	10
10.2	1.26	4.43	3.98	3.17	14.196	85.804	809.130	117.780	926.910	87.293						
10.3	1.28	5.43	4.57	4.15	20.723	79.277	806.410	117.330	923.740	87.298						
10.4	1.32	3.59	3.19	2.27	17.621	82.379	803.120	116.470	919.590	87.335						
15.1	1.25	5.57	4.66	4.32	21.065	78.935	801.250	116.070	917.320	87.347	2.2086	0.0221	0.3324	0.0020	0.0608	15
15.2	1.26	4.80	4.21	3.54	16.667	83.333	797.840	115.160	913.000	87.387						
15.3	1.28	5.87	4.93	4.59	20.479	79.521	794.890	114.570	909.460	87.402						
15.4	1.28	4.80	4.19	3.52	17.330	82.670	791.240	113.630	904.870	87.442						
20.1	1.25	4.72	4.03	3.47	19.885	80.115	788.330	113.020	901.350	87.461	2.0992	0.0210	0.3312	0.0020	0.0577	20
20.2	1.28	5.30	4.65	4.02	16.169	83.831	785.550	112.330	897.880	87.489						
20.3	1.24	3.40	2.95	2.16	20.833	79.167	782.180	111.680	893.860	87.506						
20.4	1.26	4.13	3.62	2.87	17.770	82.230	780.470	111.230	891.700	87.526						
25.1	1.27	4.75	4.17	3.48	16.667	83.333	778.110	110.720	888.830	87.543	1.6553	0.0166	0.3302	0.0020	0.0443	25
25.2	1.26	5.68	4.83	4.42	19.231	80.769	775.210	110.140	885.350	87.560						
25.3	1.26	4.02	3.45	2.76	20.652	79.348	771.640	109.290	880.930	87.594						
25.4	1.30	4.37	3.80	3.07	18.567	81.433	769.450	108.720	878.170	87.620						

Table 14: Data acquired during the filtration method for the V-blender filled to 30% continued.

V-Blender @ 30% fill																	
Sample number #	Filter weight (g)	Filter & sample (g)	After H2O (g)	Sample weight (g)	Combination (%)	Maize (%)	Maize remaining (g)	Combination remaining (g)	Total remaining (g)	Maize remaining (%)	Standard deviation (%)	Standard deviation (ratio)	Standard deviation of total mixture	Acceptable deviation limit	Mixing index	Revolution number
30.1	1.24	4.47	3.82	3.23	20.124	79.876	766.950	108.150	875.100	87.641	3.0935	0.0309	0.3291	0.0020	0.0885	30
30.2	1.25	6.54	5.81	5.29	13.800	86.200	764.370	107.500	871.870	87.670						
30.3	1.29	3.87	3.36	2.58	19.767	80.233	759.810	106.770	866.580	87.679						
30.4	1.29	3.83	3.43	2.54	15.748	84.252	757.740	106.260	864.000	87.701						
35.1	1.24	5.41	4.65	4.17	18.225	81.775	755.600	105.860	861.460	87.712	1.4950	0.0149	0.3283	0.0020	0.0397	35
35.2	1.31	5.05	4.45	3.74	16.043	83.957	752.190	105.100	857.290	87.740						
35.3	1.26	3.60	3.15	2.34	19.231	80.769	749.050	104.500	853.550	87.757						
35.4	1.23	3.78	3.36	2.55	16.471	83.529	747.160	104.050	851.210	87.776						
40.1	1.24	4.24	3.74	3.00	16.667	83.333	745.030	103.630	848.660	87.789	2.1238	0.0212	0.3274	0.0020	0.0591	40
40.2	1.28	4.55	3.87	3.27	20.795	79.205	742.530	103.130	845.660	87.805						
40.3	1.30	4.88	4.17	3.58	19.832	80.168	739.940	102.450	842.390	87.838						
40.4	1.24	3.81	3.38	2.57	16.732	83.268	737.070	101.740	838.810	87.871						
50.1	1.28	5.00	4.26	3.72	19.892	80.108	734.930	101.310	836.240	87.885	2.4311	0.0243	0.3263	0.0020	0.0688	50
50.2	1.24	4.25	3.76	3.01	16.279	83.721	731.950	100.570	832.520	87.920						
50.3	1.26	3.38	2.95	2.12	20.283	79.717	729.430	100.080	829.510	87.935						
50.4	1.29	4.44	3.95	3.15	15.556	84.444	727.740	99.650	827.390	87.956						
60.1	1.26	4.14	3.59	2.88	19.097	80.903	725.080	99.160	824.240	87.970	0.9508	0.0095	0.3253	0.0020	0.0232	60
60.2	1.26	4.18	3.67	2.92	17.466	82.534	722.750	98.610	821.360	87.994						
60.3	1.31	3.92	3.41	2.61	19.540	80.460	720.340	98.100	818.440	88.014						
60.4	1.26	4.53	3.94	3.27	18.043	81.957	718.240	97.590	815.830	88.038						
80.1	1.27	4.33	3.79	3.06	17.647	82.353	715.560	97.000	812.560	88.062	0.4047	0.0040	0.3242	0.0020	0.0064	80
80.2	1.27	4.80	4.17	3.53	17.847	82.153	713.040	96.460	809.500	88.084						
80.3	1.26	3.26	2.90	2.00	18.000	82.000	710.140	95.830	805.970	88.110						
80.4	1.27	3.96	3.46	2.69	18.587	81.413	708.500	95.470	803.970	88.125						
100.1	1.31	5.65	4.86	4.34	18.203	81.797	706.310	94.970	801.280	88.148	0.8930	0.0089	0.3232	0.0020	0.0216	100
100.2	1.26	3.83	3.41	2.57	16.342	83.658	702.760	94.180	796.940	88.182						
100.3	1.24	3.85	3.39	2.61	17.625	82.375	700.610	93.760	794.370	88.197						
100.4	1.29	4.50	3.97	3.21	16.511	83.489	698.460	93.300	791.760	88.216						

Table 15: Data acquired during the filtration method for the V-blender filled to 60%.

V-Blender @ 60% fill																
Sample number #	Filter weight (g)	Filter & sample (g)	After H2O (g)	Sample weight (g)	Combination (%)	Maize (%)	Maize remaining (g)	Combination remaining (g)	Total remaining (g)	Maize remaining (%)	Standard deviation (%)	Standard deviation (ratio)	Standard deviation of total mixture	Acceptable deviation limit	Mixing index	Revolution number
2.1	1.29	4.49	4.36	3.20	4.063	95.938	1706.94	255.060	1962.000	87.000	17.9009	0.1790	0.3363	0.0020	0.5295	2
2.2	1.27	3.73	3.57	2.46	6.504	93.496	1703.87	254.930	1958.800	86.985						
2.3	1.28	3.24	3.15	1.96	4.592	95.408	1701.57	254.770	1956.340	86.977						
2.4	1.25	4.78	3.34	3.53	40.793	59.207	1699.7	254.680	1954.380	86.969						
4.1	1.27	4.28	5.20	3.01	-30.565	130.565	1697.61	253.240	1950.850	87.019	22.5153	0.2252	0.3361	0.0020	0.6679	4
4.2	1.30	4.96	4.49	3.66	12.842	87.158	1693.68	254.160	1947.840	86.952						
4.3	1.28	4.69	4.59	3.41	2.933	97.067	1690.49	253.690	1944.180	86.951						
4.4	1.23	3.13	2.74	1.90	20.526	79.474	1687.18	253.590	1940.770	86.934						
6.1	1.28	4.93	4.85	3.65	2.192	97.808	1685.67	253.200	1938.870	86.941	10.0298	0.1003	0.3370	0.0020	0.2935	6
6.2	1.24	3.61	3.21	2.37	16.878	83.122	1682.1	253.120	1935.220	86.920						
6.3	1.28	3.70	3.67	2.42	1.240	98.760	1680.13	252.720	1932.850	86.925						
6.4	1.27	4.68	3.97	3.41	20.821	79.179	1677.74	252.690	1930.430	86.910						
8.1	1.29	4.64	4.43	3.35	6.269	93.731	1675.04	251.980	1927.020	86.924	15.9800	0.1598	0.3371	0.0020	0.4708	8
8.2	1.20	6.02	4.47	4.82	32.158	67.842	1671.9	251.770	1923.670	86.912						
8.3	1.27	3.88	3.73	2.61	5.747	94.253	1668.63	250.220	1918.850	86.960						
8.4	1.25	4.73	3.51	3.48	35.057	64.943	1666.17	250.070	1916.240	86.950						
10.1	1.27	5.73	5.57	4.46	3.587	96.413	1663.91	248.850	1912.760	86.990	14.1554	0.1416	0.3364	0.0020	0.4173	10
10.2	1.26	6.53	5.27	5.27	23.909	76.091	1659.61	248.690	1908.300	86.968						
10.3	1.31	4.16	3.97	2.85	6.667	93.333	1655.6	247.430	1903.030	86.998						
10.4	1.26	6.24	4.58	4.98	33.333	66.667	1652.94	247.240	1900.180	86.989						
15.1	1.29	4.97	4.44	3.68	14.402	85.598	1649.62	245.580	1895.200	87.042	11.8831	0.1188	0.3358	0.0020	0.3500	15
15.2	1.29	6.70	5.60	5.41	20.333	79.667	1646.47	245.050	1891.520	87.045						
15.3	1.28	6.29	6.14	5.01	2.994	97.006	1642.16	243.950	1886.110	87.066						
15.4	1.27	5.43	4.12	4.16	31.490	68.510	1637.3	243.800	1881.100	87.039						
20.1	1.27	4.46	4.21	3.19	7.837	92.163	1634.45	242.490	1876.940	87.081	12.4296	0.1243	0.3354	0.0020	0.3668	20
20.2	1.27	6.07	4.79	4.80	26.667	73.333	1631.51	242.240	1873.750	87.072						
20.3	1.27	3.96	3.78	2.69	6.691	93.309	1627.99	240.960	1868.950	87.107						
20.4	1.28	2.95	2.44	1.67	30.539	69.461	1625.48	240.780	1866.260	87.098						
25.1	1.32	4.63	4.35	3.31	8.459	91.541	1624.32	240.270	1864.590	87.114	7.9100	0.0791	0.3350	0.0020	0.2315	25
25.2	1.26	5.84	4.78	4.58	23.144	76.856	1621.29	239.990	1861.280	87.106						
25.3	1.27	3.91	3.68	2.64	8.712	91.288	1617.77	238.930	1856.700	87.131						
25.4	1.30	2.99	2.63	1.69	21.302	78.698	1615.36	238.700	1854.060	87.126						

Table 16: Data acquired during the filtration method for the V-blender filled to 60% continued.

V-Blender @ 60% fill																
Sample number #	Filter weight (g)	Filter & sample (g)	After H2O (g)	Sample weight (g)	Combination (%)	Maize (%)	Maize remaining (g)	Combination remaining (g)	Total remaining (g)	Maize remaining (%)	Standard deviation (%)	Standard deviation (ratio)	Standard deviation of total mixture	Acceptable deviation limit	Mixing index	Revolution number
30.1	1.23	5.30	4.92	4.07	9.337	90.663	1614.03	238.340	1852.370	87.133	10.9420	0.1094	0.3348	0.0020	0.3227	30
30.2	1.27	3.85	3.15	2.58	27.132	72.868	1610.34	237.960	1848.300	87.125						
30.3	1.29	4.73	4.46	3.44	7.849	92.151	1608.46	237.260	1845.720	87.145						
30.4	1.27	3.35	2.77	2.08	27.885	72.115	1605.29	236.990	1842.280	87.136						
35.1	1.26	4.55	4.18	3.29	11.246	88.754	1603.79	236.410	1840.200	87.153	7.5839	0.0758	0.3346	0.0020	0.2220	35
35.2	1.31	4.93	4.03	3.62	24.862	75.138	1600.87	236.040	1836.910	87.150						
35.3	1.24	3.59	3.34	2.35	10.638	89.362	1598.15	235.140	1833.290	87.174						
35.4	1.28	4.56	3.80	3.28	23.171	76.829	1596.05	234.890	1830.940	87.171						
40.1	1.27	5.24	4.89	3.97	8.816	91.184	1593.53	234.130	1827.660	87.190	7.1724	0.0717	0.3342	0.0020	0.2099	40
40.2	1.27	4.67	3.93	3.40	21.765	78.235	1589.91	233.780	1823.690	87.181						
40.3	1.28	4.77	4.38	3.49	11.175	88.825	1587.25	233.040	1820.290	87.198						
40.4	1.31	6.09	5.00	4.78	22.803	77.197	1584.15	232.650	1816.800	87.195						
50.1	1.25	5.13	4.68	3.88	11.598	88.402	1580.46	231.560	1812.020	87.221	7.1317	0.0713	0.3339	0.0020	0.2089	50
50.2	1.29	4.75	3.97	3.46	22.543	77.457	1577.03	231.110	1808.140	87.218						
50.3	1.28	5.79	5.47	4.51	7.095	92.905	1574.35	230.330	1804.680	87.237						
50.4	1.27	4.32	3.72	3.05	19.672	80.328	1570.16	230.010	1800.170	87.223						
60.1	1.26	4.35	3.94	3.09	13.269	86.731	1567.71	229.410	1797.120	87.235	4.2545	0.0425	0.3337	0.0020	0.1222	60
60.2	1.30	4.44	3.79	3.14	20.701	79.299	1565.03	229.000	1794.030	87.235						
60.3	1.24	4.07	3.69	2.83	13.428	86.572	1562.54	228.350	1790.890	87.249						
60.4	1.28	4.56	3.88	3.28	20.732	79.268	1560.09	227.970	1788.060	87.250						
80.1	1.29	4.86	4.40	3.57	12.885	87.115	1557.49	227.290	1784.780	87.265	3.7595	0.0376	0.3334	0.0020	0.1074	80
80.2	1.28	5.36	4.53	4.08	20.343	79.657	1554.38	226.830	1781.210	87.265						
80.3	1.23	4.21	3.78	2.98	14.430	85.570	1551.13	226.000	1777.130	87.283						
80.4	1.30	5.04	4.30	3.74	19.786	80.214	1548.58	225.570	1774.150	87.286						
100.1	1.26	3.98	3.55	2.72	15.809	84.191	1545.58	224.830	1770.410	87.301	2.1922	0.0219	0.3330	0.0020	0.0602	100
100.2	1.25	3.52	3.09	2.27	18.943	81.057	1543.29	224.400	1767.690	87.305						
100.3	1.28	3.80	3.42	2.52	15.079	84.921	1541.45	223.970	1765.420	87.314						
100.4	1.27	4.82	4.13	3.55	19.437	80.563	1539.31	223.590	1762.900	87.317						