A Kinematic Controller for Liquid Pouring between Vessels Modelled with Smoothed Particle Hydrodynamics

Gabriel Camporredondo [1,*]**, Ramón Barber** [1,*]**, Mathieu Legrand** [2] **and Lourdes Muñoz** [3]

[1] Department of Systems and Automation, Charles III University of Madrid, 28911 Madrid, Spain
[2] Department of Mechanical Chemical and Design Engineering, Technical University of Madrid,
    28040 Madrid, Spain; mathieu.legrand@upm.es
[3] Department of Computing Technologies, Monterrey Institute of Technology, CDMX 01389, Mexico;
    lmunoz-gomez@irobot.com
*   Correspondence: 100309563@alumnos.uc3m.es (G.C.); rbarber@ing.uc3m.es (R.B.);
    Tel.: +52-55-3889-9292 (G.C.); +34-91-624-8792 (R.B.)

check for updates

Featured Application: Automatic free surface liquid transfer for service robotics in domestic spaces where this kinematic model can be adapted to two manipulators, one for pouring and one for catching.

Abstract: In robotics, the task of pouring liquids into vessels in non-structured or domestic spaces is an open field of study. A real time, fluid dynamic simulation, based on smoothed particle hydrodynamics (SPH), together with solid motion kinematics, allow for a closed loop control of pouring. In the first place, a control criterion related with the behavior of the liquid free surface is established to handle sloshing, especially in the initial phase of pouring to prevent liquid adhesion over the vessel rim. A 2-D, free surface SPH simulation is implemented on a graphic processing unit (GPU) to predict the liquid motion with real-time capability. The pouring vessel has a single degree of freedom of rotation, while the catching vessel has a single degree of freedom of translation, and the control loop handles the tilting angle of the pouring vessel. In this work, a two-stage pouring method is proposed, differentiating an initial phase where sloshing is particularly relevant, and a nearly constant outflow phase. For control purposes, the free outflow trajectory was simplified and modelled as a free falling solid with an initial velocity at the vessel crest, as calculated by the SPH simulation. As the first stage of pouring is more delicate, a novel slosh induction method (SIM) is proposed to overcome spilling issues during initial tilting in full filled vessels. Both robotic control and fluid modelling showed good results at multiples initial vessel filling heights.

Keywords: robotic pouring of liquids; free surface fluid motion control; smoothed particle hydrodynamics; PID control

1. Introduction

The concept of robots performing domestic tasks is still under development [1,2]. In particular, to transfer fluids between vessels by robotic manipulators is the general scope of this research. A full robotic fluid pouring task based on physical modelling in unconstrained settings is a complex task. Vessel identification, grasping, fluid dynamics and solid body kinematics are some of the modelling tasks to perform by robotics. Coupling fluid dynamics and solid kinematics for pouring liquids is the particular subject of this work. The purpose is to provide a robust control for pouring using a free surface model that can overcome spillage regardless of the boundary conditions in tested conditions.

Fully comprehensive reasoning of pouring liquids can be found in [1]. There, the problem is described as a general ontology, where physical dynamics are set aside, and the logic of pouring is conducted so further approaches of pouring can follow the same logical principles. However, quantitative approaches are case-dependent. By applying physical governing equations and constraining them to a given geometry, space and the relevant computational limitations, any particular model will arise with new issues.

The problem of a flow over a free overfall in an open channel has been studied widely by assuming parallel streamlines of the flow and no compaction of the nappe [2,3]. This assumption turns to be very efficient for tracking the flow trajectories at the expense of fine detail visualization. On the other hand, pure fluid simulation methods have been developed to leverage the potential of novel parallel computation using GPUs to perform fast fluid simulations like the smoothed particle hydrodynamics method (SPH) [4], or the volume-of-fluid method [5]. Both methods must deal with their own limitations (i.e., boundary conditions), but both have proven consistency in free surface modelling [6,7]. Specially, in the case of SPH, fluid particles' motion properties are processed using the GPU to avoid CPU overheads [4]. SPH was implemented and discussed in the work described herein.

Recent work has developed adaptive pouring of liquids method [8,9] based on particle position instead of a free surface approach like the present work. Using a SPH flow, the method in [9] relies in the position of fluid particles. The approximation of the particle positions is supported by a calibration stage where an added discrepancy reduces the cost of spillage during pouring. The ability to classify collected versus spilled fluid is the main contribution here.

Other training methods of pouring have been developed recently [10,11]. A training-based approach using thermal imaging for feedback is used in [10]. It determines liquid volumes using neural networks and a proportional-integral-derivative (PID) controller for robot kinematics. Neural networks eventually have to be reinforced with new training when pouring phases switch between fluid regimes. Nonetheless, the usage of a PID controller is interesting for our work because it shares the idea of controlling the final position of the free fall. In our work, the PID controls the fluid free surface rather than only the vessel geometry, as in [10]. The work in [11] trains the pouring task by learning imitations. Fluid dynamics are not part of the method; rather, a trial and error method is used by the robot to learn from human demonstrations.

Typically, rigid body and fluid motion models can be merged together in a closed loop controller to transfer liquid from one vessel to the other at constant outflow velocity with relative ease using free falling parabolic trajectories [12]. However, maintaining constant outflow requires a dedicated control, and moreover, initial and sometimes final phases of pouring are inherently non stationary. Another challenge to consider is the issue of liquid adhesion to the vessel crest. Usually, this is avoided by using a sharp spilling edge and/or guaranteeing a minimum velocity at the outlet edge. That is why in this work, the outflow was controlled by maintaining a constant liquid elevation over the vessel's crest, which is especially relevant during the initial surface oscillation. Grundelius in [13] provides an explanation of how to minimize the slosh during rotational and translational motion. In fact, the slosh control described in [13] was used here to control the height of liquid over the crest.

Section 2 describes the fluid model used (SPH) and its implementation in parallel processing using GPU libraries. Section 3 describes the two models involved: the solid vessel kinematic model and the free surface model. A model of slosh induction for the pouring task is also described. Section 4 describes and illustrates the results, and a detailed discussion is presented. Section 5 provides conclusions and future work.

2. Fluid Models

The goal of pouring liquid is to estimate the final spatial location where the flow will drop in, at a certain pouring height and with a certain vessel motion.

Considering a hydrostatic equilibrium with an orthogonal free surface, a fluid discharge in an open space can be simplified as a free fall of solid motion dynamics in stationary conditions [14].

However, this is only true if outflow velocity remains within a valid range of laminar flow velocities. Case a in Figure 1 illustrates the free fall fluid motion that is to be aimed for with predictable curvature over time.
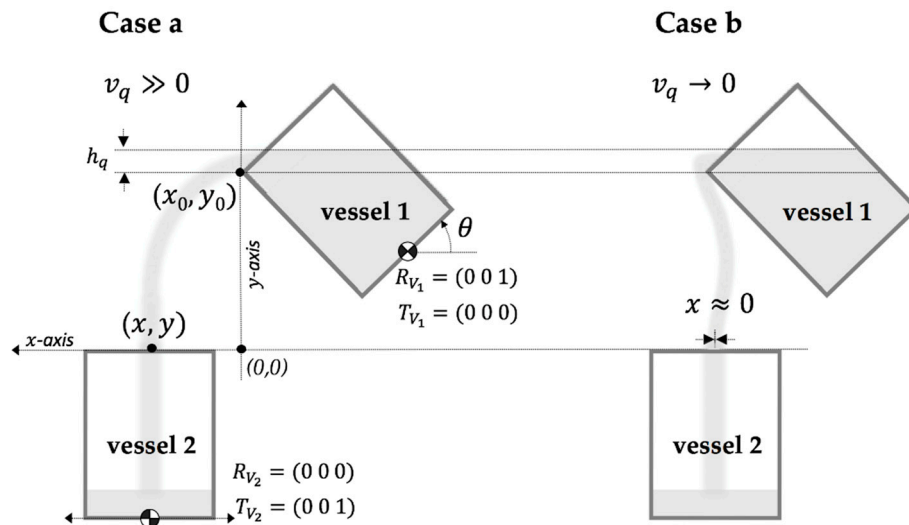

Figure 1. Case a. Expected laminar flow with free fall trajectory slope. Case b. Unexpected no-slope turbulent fall trajectory.

In contrast, unpredictable flow motion in fluid free falls may happen in a fluid with a free surface in motion (not orthogonal to the gravity vector). See Figure 1 case b. Even when initial conditions of hydrostatic equilibrium are met, turbulent outflow can occur during pouring when flow velocity, v_q, remains particularly low. The most typical causes of turbulence at low v_q values are the border shape of the vessel (Coanda effect) and the force of surface tension.

In this work, we desired to address the most visible source of turbulence: the turbulence due to low outflow velocities. To overcome this turbulence, the pouring task was divided into two stages to separate the low v_q flows from the high v_q flows. This work also contributes a new method of slosh induction to link these two stages with a control action. SIM uses the flow velocity of an induced slosh to produce replicable laminar flow at different tilting angles during initial pouring. Section 4 describes this method and its kinematic phases.

Using dimensional analysis, that which may occur when Weber number $We = 2\rho g h_q^2/3\sigma$ is small (ρ is water density; g, gravity acceleration; h_q, the liquid film height over the vessel rim; and σ, the surface tension of the fluid). For water, the condition for avoiding rim-attached flow roughly corresponds to $h_q > 3$ mm.

A bimanipulated liquid transfer kinematic model is proposed in our simulations. One vessel for pouring and one for catching. Vessel 1, the vessel of pouring, will only have a rotational motion, and vessel 2, the vessel of catching, will only have a translational motion; see Figure 1. A pouring height measured from the vessel 1 tip to the top border of vessel 2 is assumed. As the aim of this work was having a real time simulation to control the robotic pouring, a 2D model was preferred to 3D because its computational cost (time) is much lower. In addition, as the simulation plane was chosen to be perpendicular to the tilting axis of the pouring vessel, we anticipated no 3D flow effects.

Modelling the fluid statics is one of the most important initial conditions to consider prior to the actual modelling of pouring motion itself. For this reason, first, the two models that set the simulation fundamentals were needed in detail. The Lagrangian fluid dynamics model is described in Section 2.1, as is the detail of its implementation to work as a free surface flow model. The second model's rigid body dynamics and its coupling with fluid are described in Section 3.

2.1. SPH Model

The SPH framework followed in this work was based on a parallel processing, just as in [15]. Its pseudocode for a single timestep cycle is illustrated in Figure 2. OpenGL library and its Utility Toolkit (GLUT) [16,17] were used to handle graphics, and OpenCL library [18] to handle GPU scripting that was processed by a graphics board (NVIDIA GFORCE GTX660ti with 1344 GPU cores) in this work.
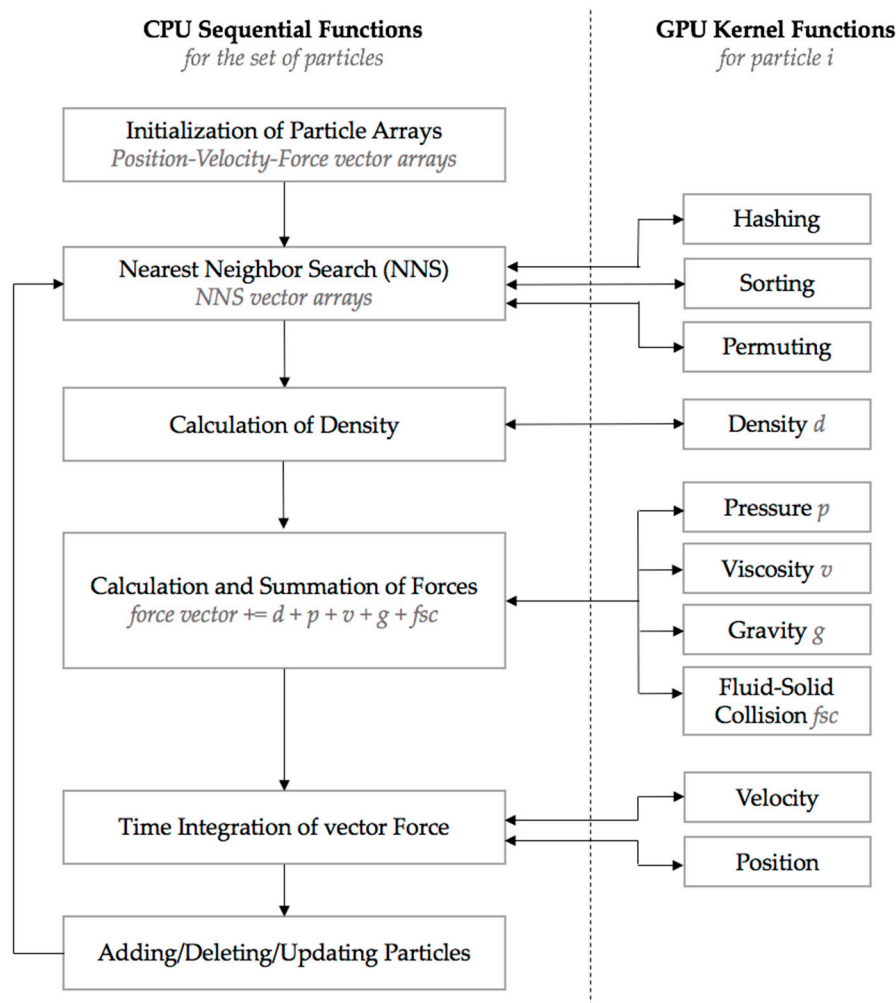

Figure 2. One time-step CPU-GPU implementation of smoothed particle hydrodynamics (SPH) fluid motion. To the left are the sequential tasks processed by the CPU. To the right are the parallel tasks (kernels) processed by the GPU.

The left part in Figure 2 is the sequential main loop processed by the CPU, and in the right part, the parallel functions (kernels) processed by the GPU. Particle data in the CPU is stored in three *N*-sized vector arrays: position, velocity and force. *N* is the number of particles running in the fluid simulation. The CPU functions form a number *N* of GPU kernels, passing them into the vector arrays to compute data in parallel.

Each CPU function block has to wait for output data from the former block so parallel processing is implemented only for each CPU function block. After vector array initialization, a nearest neighbor search algorithm (NNS) [15] is needed to provide the surrounding particles that will contribute to the force acting on particle *i*. Initially, in NNS, the 2D domain is divided into cells using a uniform grid so a cell index can store each particle where it belongs. Then, a spatial hash value is created using the previous cell index, and particles are sorted according to this hash value. A bi-tonic sorting algorithm is used to take advantage of the GPU processing [19]. Finally, the vector arrays of position,

velocity and force are permuted according to their sorted hash values. The NSS at the end will provide a space map on which particles are located so that finding neighbors can be iterated as quickly and straightforwardly as possible.

The less time-consuming the NNS algorithm is, the more helpful to the next parts of the SPH implementation it will be. This is why GPU processing turns out to be fundamental to computing the hashing, the sorting and the permuting. OpenCL threads can be synchronized to optimize global and local GPU memory transfers. The implementations of this work are available from NVIDIA [19] and [20].

The next block in Figure 2 is the calculation of liquid density from the individual SPH particles. Equation (1) defines density as the sum of masses weighted by a *W* function. *W* is the interpolation kernel that computes the position of particles *i* and *j* within the distance *h*, also defined as the smoothing distance by SPH formalisms. A cubic spline function is used for *W*; this provides compact support in its derivatives, just as in [7], and efficiently smooths away the acting neighboring forces with a relatively minor computational cost. Density is the first to compute because it is needed by the next parallel computed forces; see Figure 2.

$$\rho_i = \sum_j m_i W(\mathbf{r}_{ij}, h). \tag{1}$$

After processing density for every fluid particle using Equation (1), a force vector for each particle is computed for pressure, viscosity and external (gravity and collisions) force contributions. Pressure *p* and viscosity *v* come from the SPH interpolation formulation, just as in [21]; see the Equations (3) and (4) used. Using the same *W* function as in Equation (1) but with first and second derivative order, the gravity force is a *y*-axis constant vector [0 9.81 0] m/s^2.

$$\frac{dV_i}{dt} = \left(f_i^{pressure} + f_i^{viscosity}\right) + g + f_i^{collision}, \tag{2}$$

$$f_i^{pressure} = -\sum_i m_i \left[\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right] \nabla_i W(\mathbf{r}_{ij}, h), \tag{3}$$

$$f_i^{viscosity} = -\mu \sum_j m_i \left[\frac{v_i + v_j}{\rho_i}\right] \nabla_i^2 W(\mathbf{r}_{ij}, h). \tag{4}$$

The pressure field is computed directly from an equation of state using weak compressibility: "Tait's equation." See [6] and [22] for reference. Equation (5) is Tait's equation, where $\rho_0 = 1000$ Kg/m^3 is the rest density of water, $c_s = 1480$ m/s is the speed of sound and $\gamma = 7$ is a selected constant. Weak compressibility is used, rather than enforcing incompressibility with the Poisson equation [7] and compromising the real time performance.

$$p = \frac{\rho_0 c_s^2}{\gamma}\left(\left(\frac{\rho}{\rho_0}\right)^\gamma - 1\right) \tag{5}$$

Finally, solving force Equations (3) and (4) for a single particle and adding them with the gravity vector *g* and the collision fluid-to-solid force, the total force of a particle *i* at certain time step is obtained and described in Equation (2). The collision force is the force acting in every particle within the range of influence of the solid boundaries (the vessel walls). Just as particles are the primitive elements of the fluid, triangles are the primitives of the solids (vessels). Therefore, fluid-to-solid collisions are particle-to-triangle collisions that are computed by trivial intersections just as in [23], with real time GPU performance.

Thus, the final computed force, *force vector* in Figure 2, is then time-integrated twice to find the updated velocity and position of particle *i* and complete one timestep computation of SPH. The leapfrog scheme is used for the integration task because of its symplectic nature; i.e., its time reversibility feature [24]. An integration time step of 2 milliseconds should be used.

Boundary conditions are far from being trivial to any control model of fluid handling. Wall boundary conditions are simulated here as collision forces between solid body primitives (triangles) and the particles. This combination has been used frequently in 3D modelling to improve processing cost at the expense of boundary stability [23]. Modelling solids as fixed particles improves stability, but also increases the processing cost by adding to the number of particles in the region to compute [25].

Boundary forces occurring in free surface interface between liquid and air (i.e., atmospheric pressure) are typically negligible for robot kinematics in domestic settings simulations.

An important parameter is the choice of the pouring velocity, or the height of liquid over the vessel rim. The faster the pour is, the faster SPH needs to calculate fluid particles at shorter time steps. But also, too low a h_q will compromise the outflow trajectory due to the fluid viscosity [26]. There is also a subcritical regime to take care of when the height increases drastically and then excites hydraulic jumps [3]. Thus, a minimal expression of h_q has to be studied based in the minimal expression of the SPH method. And, this is the particle itself.

Each particle has a range of influence. That range, called smoothing length in SPH (h in Equations (3) and (4)), is defined by a spatial range where neighboring particles influence a given particle. By defining the minimum size of h_q as the size of the smoothing length, the particles carry along the flow of the added forces of the particles around it. This diminishes the sources of turbulence and provide added forces to the outflow every time step. In our simulation, a scaled smoothing length of 10 mm was used, and thus a radius of interaction of 5 mm as the minimum value for h_q.

2.2. Free Fall Model

In this work, a typical free fall curvature with free nappe and sharp-crest type outflow was used, just like in Figure 1 case a. When leaving the vessel, the characteristic dimension of the liquid jet ($\sim h_q$) is generally small so that neither boundary conditions nor fluid intrinsic forces have great contributions to open space free fall. Under these assumptions, fluid particles describe a quadratic trajectory defined by Equations (6) and (7).

$$y = y_0(t) - \frac{1}{2}gt^2, \tag{6}$$

$$x = x_0(t) + v_{o_x}t. \tag{7}$$

During initial pouring, the velocity in the y-axis equals zero; the initial flow position y_0 gradually changes in time; and the catching vessel 2 remains at $y = 0$. The time t at which the fluid particles enter vessel 2 can be calculated with Equation (6). Assuming null horizontal acceleration, the x position of catching (when $y = 0$) in Equation (7) reduces to $x = v_{o_x}t$, where $v_{o_x} = v_q$ is the averaged bulk velocity at the vessel outlet. This velocity could be determined analytically as $v_q = \frac{2}{3}C_d\sqrt{2gh_q}$, where C_d is an empirical discharge coefficient $\sim$0,6. However, v_q is obtained directly from averaging the SPH particle velocities in the outflow plane at each time step of the simulation. Then, x of catching, in Equation (7), is used to calculate the ending x-axis position at which vessel 2 should be translated to.

This is true for a pouring stage with constant outflow, called the constant flow stage (CFS) hereafter; however, the initial pouring stage imposes an increasing and decreasing outflow at the vessel tip, and thus, an incremental x-acceleration. Therefore, it requires us to divide the pouring task into two stages. In Section 3, a two-stage pouring task, its modelling and its control solution are described.

3. Modelling and Control

A pouring method with two stages based on fluid velocity at the vessel tip is defined. One initial stage called the approaching flow stage, (AFS), happens when the vessel initiates tilting to reach a window level with height h_q (see Figure 3) and a second stage, the CFS, where liquid flows at constant rate out of the pouring vessel, keeping h_q constant by a control action. The next two sections describe AFS and CFS stages in detail. Final comments in this section will describe the relevance of pouring over catching due to a control action in AFS and CFS.
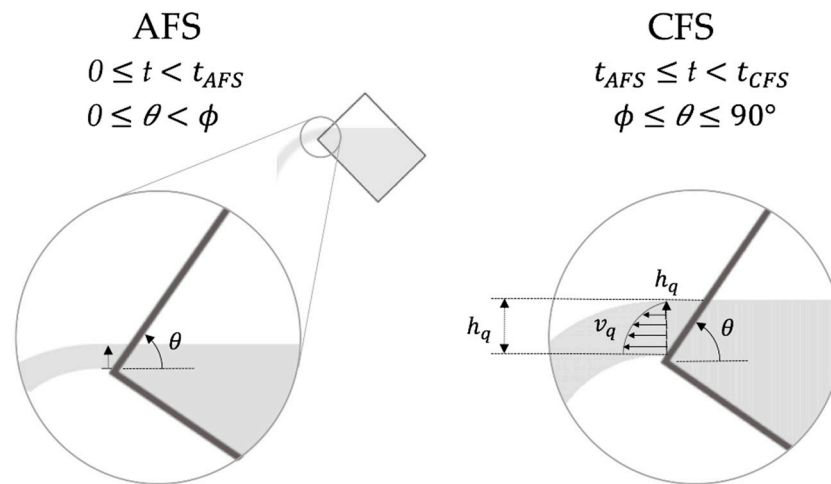
Figure 3. Stages of pouring. To the (**left**), the approaching flow stage (AFS) going from the initial time to t_{AFS}. To the (**right**), the succeeding stage of constant flow (CFS) going from t_{AFS} to t_{CFS}.

3.1. Approaching Flow Stage (AFS)

In this initial stage, tilting angle θ goes from zero to a threshold value of ϕ, so the stabilized outflow height reaches a threshold value h_q at a certain outflow bulk velocity v_q; see Figure 3.

While for a high v_q, the outflow follows a free falling motion, the flow may attach to the walls of the vessel at low values of v_q and θ, causing liquid to drip through the vessel walls; see Figure 1 case b. Regarding other factors that may be involved in this kind of flow, however, most of them can be prevented by guaranteeing a minimum value of v_q at the vessel tip.

To overcome this initial issue, inspired by intuitive human pouring action, a slosh induction method (SIM) is proposed in this work. SIM consists of creating an anti-symmetrical oscillation of the fluid free surface such that the tilting angle θ of the rotated vessel 1 produces an equal angle θ_1 of the liquid-free surface at a certain liquid level condition. This condition is set when the liquid level h_e is equal or lower than the window level h_q at an arbitrary angle of $\theta = 20°$. See Figure 4 for reference.
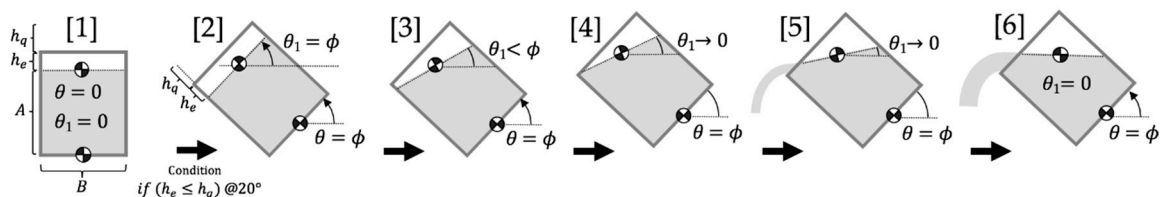

Figure 4. Transitions of AFS with SIM. Transition 1 defines initial conditions in hydrostatic equilibrium where the angle of the pouring vessel and the angle of the free surface are zero. Transitions 2–6 define the slosh induction method steps that run only if the height condition ($h_e \leq h_q$ at 20°) is met.

SIM will not produce a significant change in the liquid volume shape from the moment of rotation of θ till the angle reaches the conditional ϕ value, leaving a free surface inclination of ϕ degrees with respect to horizontal, such as it is indicated in transition 1–2 of Figure 4. That retains a linear model for the slosh. After θ_1 and θ become ϕ, free surface will freely return to the flat horizontal shape (see Figure 4 transitions 3–6) allowing the slosh to travel back to the opposite vessel wall and run out of the vessel with a velocity close to the fundamental velocity of the induced slosh.

When the liquid level condition is not met, there is no need to apply AFS control and CFS can be resumed. In such a case, the outflow will follow free fall motion from the 0° and until the final angle, the x-axis position of the fluid fall is predictable, since the x-component velocity of flow (v_q) is known by SPH computations. This is one advantage of using the SPH method as an emulator engine.

Lee [27] studied the moment of inertia of liquid in a vessel, and showed that, as the area of liquid gets more symmetrical, the moment of inertia becomes smaller, and, in the same way, the portion of liquid that moves along the vessel becomes smaller. Therefore, as the moment of inertia states, without using SIM, the location of the center of rotation would be right at the center of mass, making the moment of inertia close to zero and retaining the free surface orthogonal to the gravity vector in such case.

In contrast, AFS with SIM uses the *z*-axis as the axis of rotation located at the middle bottom of the vessel 1. This increases the moment of inertia to allow the free surface to elevate along the vessel bottom plane. See Figure 4 transition 2.

In this work, AFS without SIM Case is not going to be developed; however, there are two steps required to run such case: (1) find the center of mass to locate the center of rotation, and (2) rotate the vessel over this center, and reach the proposed outflow height h_q. Recall that in this case, one could have a scenario of a brief volume to pour with respect to the total vessel volume, and thus, CFS could not begin with the initial condition of h_q. This is important to note, because the selection of h_q has to be always minimal (close to the smoothing length h of SPH), specially at low pouring volumes.

This rationale of the AFS needs to be developed in a mathematical model such that the kinematics can be useful for slosh control. One approach used to find this model is to resolve the boundary condition equations from the velocity potential of the flow [28]. This will lead to a nonlinear and continuous formulation of the slosh that will demand a high computational cost even for a single degree of freedom of robotic kinematics.

Besides being nonphysical, if it is only considered the free surface as a function of θ, its velocity and acceleration functions would lead to a high, non-linear equation to be eventually solved with the control. Instead of this, a linear model is preferred, as is applied in different modelling schemes [13,28,29] and that has proven equivalency for small oscillation amplitudes. This model describes the slosh in a free surface as a finite number of oscillators using poorly damped, second order pendulums with less energy dissipation. Each pendulum describes each oscillation component within the slosh.

In our case, this model fits well since it only needs to find the velocity of the fundamental slosh component when SIM is used in AFS. In the graph of [30] (pp. 134–135), it is clear how the fundamental component contributes to more than the 80% of the total slosh energy, so processing the rest of the slosh components should not add a significant magnitude to the slosh velocity. Further details on this model can be found in [29]. The pendulum model does not consider flow viscosity; that plays a minor role in free surface oscillation, but it has shown to be an accurate model [29,30]. In addition, its predictions match the SPH simulation of the present work.

Figure 5 diagram describes this model, where L_0, m_0 and I_0 describe the height, the mass and moment of inertia of the rigid model; L_n, l_n, m_n and θ_n are the height, the pendulum distance, the mass and the angle of the *n*th pendular component—*n = 1* being the fundamental pendular component.
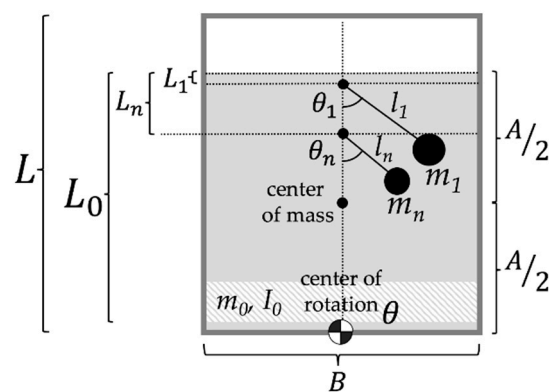

Figure 5. Pendulum model of the slosh, including *n*th oscillations of the free surface.

Notice that θ is the tilting angle of the vessel while θ_n values are the rotation angles of the slosh components represented by n pendulums. g is the gravity force and A the liquid depth. I_0 is the moment of inertia of the liquid volume; however, by solving Lagrange's equation for lateral (x) and rotational (θ) excitations, Ibrahim in [29], ends having a slosh equation that is independent of I_0 for rectangular and cylindrical vessels. Equation (8) shows the slosh equation of this linear model coming from this reference.

$$\sum_{n=1}^{\infty} m_n l_n \left[\left(\frac{A}{2} - L_n - l_n \right) \ddot{\theta} + l_n \ddot{\theta}_n + \ddot{x} \right] + \sum_{n=1}^{\infty} m_n g l_n (\theta + \theta_n) = 0. \tag{8}$$

Notice that Equation (8) will give a linear relationship between θ and θ_n. This linearity is typically achieved by approaching the sine function of θ of the native pendulum equation to the single variable of θ when the angular displacements are small. Additionally, notice that Equation (8) lacks angular velocity $\dot{\theta}_n$, meaning that it corresponds to an undamped time response of the slosh.

If our interest is only the fundamental oscillation of the slosh and no lateral excitation will be needed ($\ddot{x} = 0$), making $n = 1$, the slosh Equation (8) yields,

$$-l_1 \ddot{\theta}_1 - g \theta_1 = \left(\frac{A}{2} - L_1 - l_1 \right) \ddot{\theta} + g \theta. \tag{9}$$

From Equation (9) and using [29], if the pendulum of interest is at $n = 1$, the distances L_n and l_n, needed to complete the time response are calculated with Equations (10) and (11).

$$l_1 = \frac{L}{\pi} \coth\left(\frac{\pi A}{L} \right), \tag{10}$$

$$L_1 = -\left[l_1 + \frac{A}{2} - \left(\frac{2L}{\pi} \tanh\left(\frac{\pi A}{L} \right) \right) \right]. \tag{11}$$

Considering an input signal of $\theta(t) = \phi \sin(\alpha t)$ to excite the vessel motion, the Equation (9) solution yields,

$$\ddot{\theta}_1 + \frac{g}{l_1} \theta_1 = \left[\left(\left(\frac{A}{2} - L_1 - l_1 \right) \frac{\alpha^2 \phi}{l_1} \right) + \frac{g \phi}{l_1} \right] \sin(\alpha t). \tag{12}$$

The characteristic polynomial of the differential Equation (12) is in form of: $s^2 + \omega^2$. s is the laplace variable, and $\omega^2 = \frac{g}{l_1}$ is the corresponding frequency equation for a simple pendulum.

$$\theta_1(t) = \frac{\phi \sin(\alpha t)}{\left(\frac{g}{l_1} - \alpha^2 \right)} \left[\left(\left(\frac{A}{2} - L_1 - l_1 \right) \frac{\alpha^2}{l_1} \right) - \frac{g}{l_1} \right]. \tag{13}$$

As Ibrahim states in [29], the slosh time response solution of (12), expressed in Equation (13), will let us compare analytically, the angle elevation of θ_1 against θ. On the right side of Figure 6, $\theta_1(t)$ is plotted for values in the range of $\pi < \alpha < 2\pi$. The dark plot is the input signal $\theta(t) = \phi \sin(\alpha t)$. It is observed here that the free surface angle $\theta_1(t)$ tends to increase the elevation between angles as α increases and until α reaches the resonant frequency $\left(\frac{g}{l_1} \right)^{1/2}$. It is clear now that this elevation will increase a non-linear and unwanted behavior of the slosh. On the left side of Figure 6, a valid range for α is used for several plots of $\theta_1(t)$. Notice that the elevation here is marginal, keeping $\alpha \ll \frac{g}{l_1}$. This will retain a linear model of the slosh, and it will make both angles of rotation, θ_1 and θ, change alike.
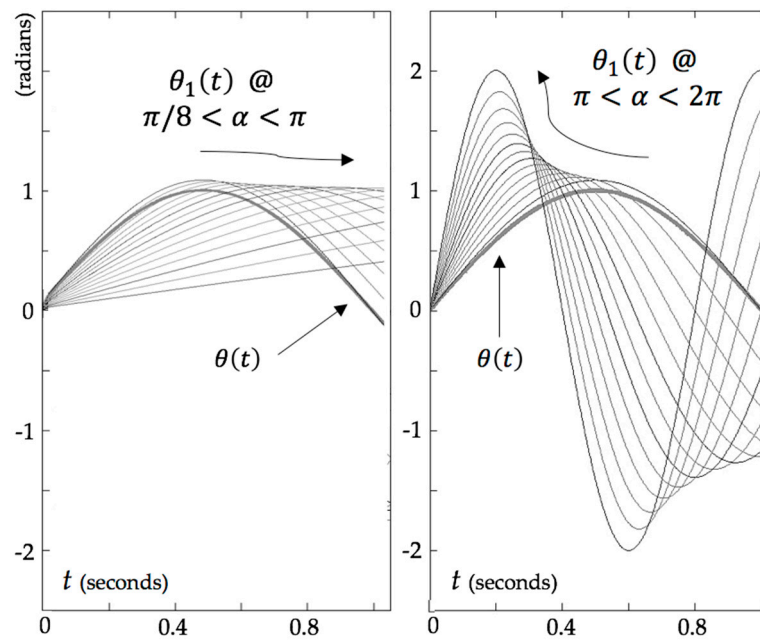
Figure 6. θ_1 transient time response for an excitation of $\theta(t) = \theta_0 \sin(\alpha t)$. Low frequency values of α to the (**left**), and high frequencies values of α to the (**right**).

The results in Section 4, illustrate the equivalence of this pendulum model of the slosh with respect to the time responses in Figure 6.

As concluded earlier in the graphs of Figure 6, a valid range of α can be obtained in an input elevation signal of θ so the elevation of θ_1 retains linearity, and ASF with SIM provides a back elevation of the slosh with equivalent angle value of θ. Figure 7 shows the proposed timeline response of ASF with SIM for the elevation of the angles.
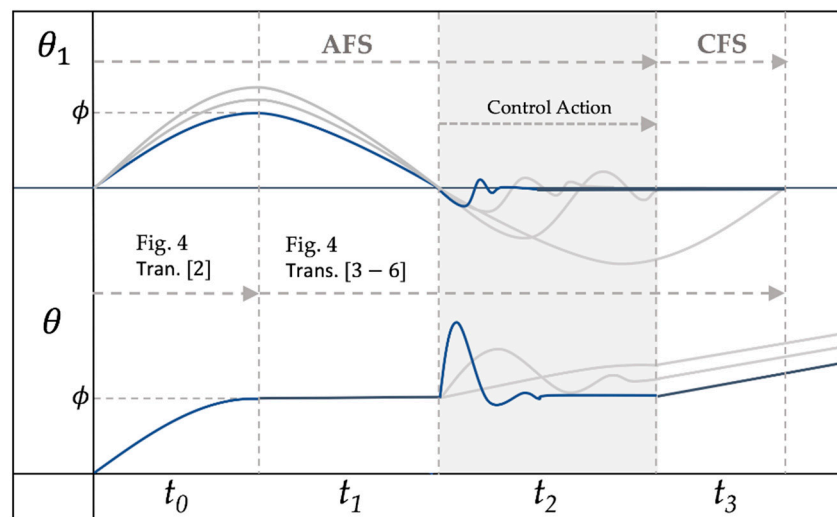

Figure 7. Proposed timeline of slosh induction method (SIM) pouring. AFS ranges from the initial tilting to the end of the control action. CFS initiates at the end of AFS and continues till the end of pouring.

In Figure 7, t_0 is the period of time where the mentioned input signal of $\theta(t)$ is applied to vessel 1 allowing the free surface angle θ_1 to turn at the same rotational velocity as θ and equally reach angle ϕ. This time period corresponds to transition 1 in Figure 4. In the next period of time, t_1, θ_1 returns to zero, while θ remains at that initial conditional angle ϕ. As soon as θ_1 crosses zero, the next period

of time, t_2, initiates with the control action of the slosh. The goal of this control is to turn the surface elevation to zero and to avoid pure undamped behavior of the slosh. Recall that an oscillating response of the slosh will be presented if no control action is taken in the pendulum model. The optimal control action would lead θ_1 to zero as soon as possible so the particle outflow velocity become constant at fixed h_q, allowing it to migrate from AFS to CFS with a stable pouring.

At the beginning of t_3, where the slosh finds free exit to flow out of the vessel, AFS will end and CFS will begin.

$$G_p = STF(s) = \frac{\theta_1(s)}{\theta(s)} = \frac{\frac{b_2}{l_1}s^2 + \frac{g}{l_1}}{s^2 + \frac{g}{l_1}},$$
$$b_2 = \left(\frac{A}{2} - L_1 - l_1\right). \tag{14}$$

A control action proposed here is based on a proportional-integral-derivative (PID) controller using the system transfer function (STF) given by Equation (14) as plant gain (G_p). This STF comes from Equation (9) where the input is the angle of vessel 1, θ, and the output, the angle of the free surface θ_1. PID control is well known for the ability to remove a steady state error, which via Equation (14), turned to be one half. It was also deduced by Equation (14) that roots exist only in the imaginary plane for an undamped time response system, so, a trained proportional-integral-derivative (PID) controller/compensator in a closed loop with G_p can add the additional zeros to dissolve the pure undamped system and turn it into a stable damped one. Using the manual tuning root locus in *Matlab*®, PID constants that better suit the damping needed to the particular system were built. Recall that the STF equation depends on the vessel dimensions, so further experimentation would require a range of PID constants for a range of vessel dimensions. In contrast, this STF will work for any filling height (A). Figure 8 shows close loop PID control block diagram. G_p is populated with the vessel dimension data used in our simulations: $A = 0.159$, $L_1 = 0.0207$, $l_1 = 0.0615$. Those values are in meters.
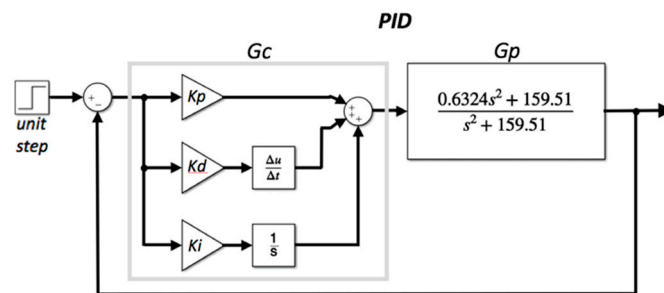

Figure 8. Simulink® block diagram of the close-loop proportional-integral-derivative (PID) control. G_p is the plant gain with populated values, and G_c is the controller gain properly tuned to reduce undamping.

The compensator gain, G_c, described by the PID controller, will add two zeros and one pole to a second order G_p. The idea is to add a dominant pole to the close loop transfer function so that the system becomes stable, damped and with minimum steady state error. The new two additional zeros will affect the amplitude of the system gain, but not the nature of the response. Equation (15) shows the PID compensator gain formula.

$$G_c = \frac{K_d s^2 + K_p s + K_i}{s}. \tag{15}$$

In Section 4, the results of running simulations are presented, including the steady state error from the close loop system and the way PID control tuning reverts the undamped response. Additionally, the numerical constants (K_p, K_i, and K_d) of the PID control were tested and are discussed in next Section.

3.2. Constant Flow Stage (CFS)

Once AFS ends at the proposed outflow height h_q, a constant flow initiates by controlling θ proportionally so the reference value h_q remains constant. This sets a steady flow at x_q. As tilting progresses, h_q will eventually drop down, even though θ increases by the control action.

CFS is proposed to pour the largest amount of liquid volume since constant outflow can be computed with less discrete error than in a variable outflow, like in the initial stage.

A volume limit is set to pour so the largest amount of poured liquid remains in this constant flow stage rather than in AFS. To accomplish this, a window level h_q should be as short as possible, so as to just clear the SPH smoothing length tolerance described earlier, in Section 2.1.

Now that h_q has been reached, CFS is a straightforward stage to process. Tilting the vessel 1 can be developed by controlling the angle θ of vessel 1 as a proportional function of the linear decrease of h_q in time (Δh_q).

$$\theta_V(t) = arctan\left(\frac{2}{B}(h_q(t) + h_e)\right). \tag{16}$$

One can go directly on with Equation (16): the trigonometric relation of the angle and the vessel 1 dimensions. However, this will give an angular step not a linear step in different center of rotation, so, instead, as Figure 9 illustrates, first measure a Δh_q at certain timestep of t_3; then, assume this height as the magnitude of a vertical vector (vector **b**) with coordinates on the tip of pouring of vessel 1. Then, calculate vector **c** from the difference of vector **a** and vector **b**. Vector **a** is the vector going from the center of rotation to the tip of pouring in vessel 1.
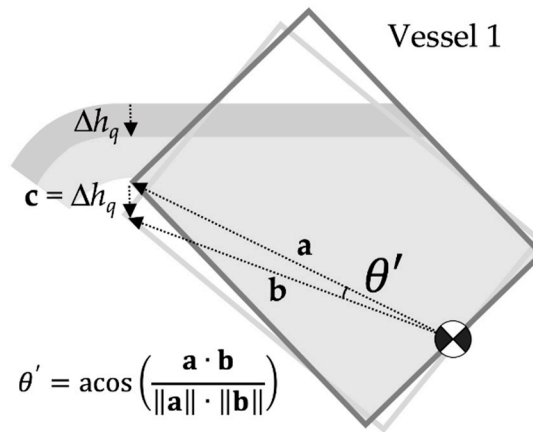

Figure 9. Vertical step of tilting in constant flow stage.

Finally, the angle θ' between the vectors **a** and **c** is calculated, and vessel 1 tilts at that angle in constant angular steps.

In simulations, a pouring for full liquid volume was developed. Therefore, h_q will drop down as the liquid volume ceases. As will be mentioned in next section, the free fall curvature has no flaws over the course of the CFS ending. It has been tried different functions in order to handle the angle of pouring not only as a function of h_q but also as a function of time with relative success. That is why [3] justifies the free fall motion of fluids as a measuring device for flow. Partial pouring would be an interesting simulation job that could be tried out in further work.

After AFS and CFS motions in vessel 1 approximate to the prospective free fall curvature, the catching motion of vessel 2 is summarized to the *x*-travel of the curvature motion. No special treatment to the catching task is required, but only to avoid large motion steps during filling to prevent spilling due to sloshing. In our simulations, the motion's update of solids takes place with each update of the particle's position (see Figure 2); therefore, there is a smoothed motion of vessel 2 along x axis.

4. Results and Discussion

First of all, a comparison of the outflow trajectory is presented between the analytical solution of Equations (6) and (7) (in red in Figure 10) and the result of the SPH simulation. As seen in Figure 10, analytical curves are fairly well-followed by the outflow particles at certain v_q values. The fall of particles at *x*-axis position along all the *y*-axis was validated. This allowed us not only to confirm a quadratic outflow curvature of the SPH method on free surface using our simulation context, but also to confirm how an outflow target during CFS can possibly be determined and tracked.
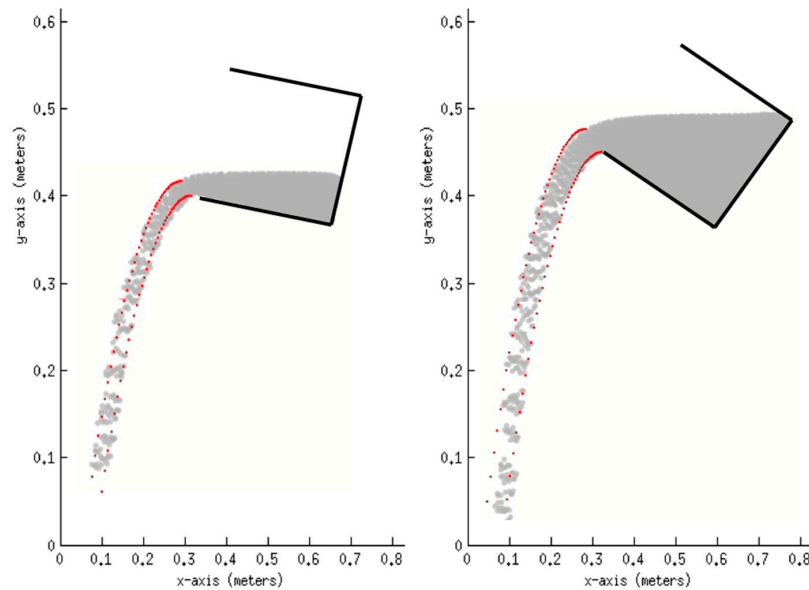

Figure 10. SPH outflows (grey) versus analytic curvatures (red). A narrow outflow to the (**left**). A broad outflow to the (**right**).

For all the following test and results, the same vessel's dimensions were used: $B = L = 0.191$ m. Unless indicated to be different, please also assume a liquid level height $A = 0.1591$ m.

Because forces, velocities and positions of particles based in a discrete time are computed, speed calibration of the system for further experimentations is another element to consider for testing. It is crucial to synchronize the main simulation loop time with the real fluid speed. This adjustment is conducted by setting the number of system loop calls so that the simulated frequency of the free surface slosh approximates to theoretical slosh frequency in horizontal motion.

For a squared vessel, a robust analytical solution to the free surface slosh frequency of the first oscillation component can be found in [13] (p. 59), and is shown in Equation (17), where *Freq* is the oscillation frequency in hertz of a free surface elevation cycle in horizontal motion, *B* is the vessel width and *A* is the fluid height at rest. It was mentioned earlier that the slosh frequency was $\omega^2 = \frac{g}{l_1}$ for the linear, pendulum-based model used; however, Equation (17) is more appropriate to use, since this computes the liquid height, that, at low heights, approximates better to the real oscillation values. Additionally, notice that in Equation (17) *Freq* is not a function of the slosh height, so the slosh induction method can compute its fundamental slosh motion regardless of the angle θ—an ideal condition to perform real time synchronization in the simulator.

$$Freq = \frac{1}{2\pi} \sqrt{\frac{g\pi}{B} \tan h\left(\frac{A\pi}{B}\right)}. \tag{17}$$

Figure 11 shows a complete free surface cycle for three different values of *A* in our simulation results. Equation (17) is used as reference to set the system time and find similar values for the experimental surface elevation cycle adjusting the SPH fluid forces.
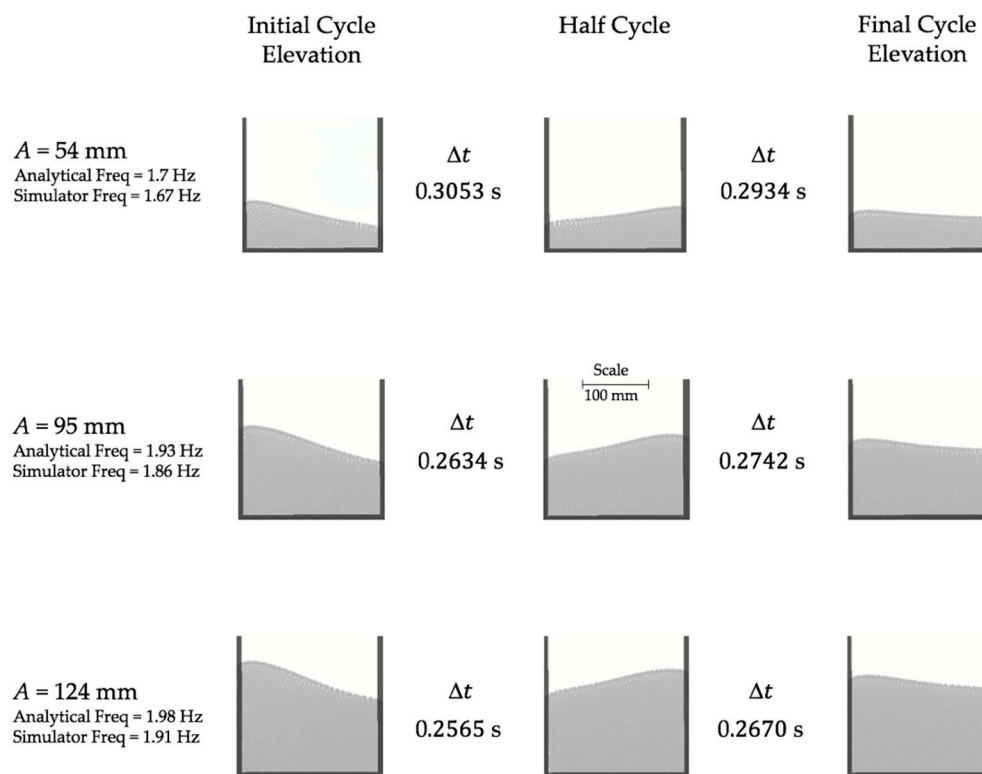
	Initial Cycle Elevation		Half Cycle		Final Cycle Elevation
A = 54 mm Analytical Freq = 1.7 Hz Simulator Freq = 1.67 Hz		Δt 0.3053 s		Δt 0.2934 s	
A = 95 mm Analytical Freq = 1.93 Hz Simulator Freq = 1.86 Hz		Δt 0.2634 s		Δt 0.2742 s	
A = 124 mm Analytical Freq = 1.98 Hz Simulator Freq = 1.91 Hz		Δt 0.2565 s		Δt 0.2670 s	

Scale: 100 mm

Figure 11. Free surface elevation cycle calibration.

Note that, as A increases, the difference between analytical and simulated values of *Freq* barely diverges. These results confirm the maximum Δt choice to run real time simulations, although a smaller Δt would give more accurate results.

As mentioned earlier, AFS without SIM is not processed. However, for that case, a center of rotation of θ would have to equal the center of mass of the liquid area. This setup cancels the moment of inertia by having equal parts of liquid mass around the center of rotation. However, there should be a maximum value of the tilting angle velocity $\dot{\theta}_{max}$ of vessel 1 at which this angle is limited to avoid spilling. Figure 12 shows the simulated values of the relation $(\theta/(27 - \theta_1))$ between the angles of rotation (θ of the vessel, and θ_1 of the slosh) versus the vessel frequency $\dot{\theta}$. From a physics perspective, it is obvious that the free surface angle will remain horizontal, with its center of rotation equal to the center of mass, and with the vertical component of $\ddot{\theta}$ below the gravity force. In Figure 12, a maximum finding of $\dot{\theta}_{max}$ is given by our running simulations. See that at the usable tilting velocities of $\dot{\theta}$, no great surface elevation happens, but only a few random disturbances due to uneven particle distribution in time.

For the case of AFS with SIM, an approach of translating the center of rotation is followed. As the center of rotation moves along the x-axis, an early elevation of the slosh due to the centrifugal force can be found. This is the principle in the SIM method: to elevate the free surface over the back wall in a linear model fashion (semi-flat surface) so the natural frequency, slosh-free return leads to an overflow on the front wall of the rotated vessel 1 (Figure 4).

As mentioned earlier, SIM has to be used when liquid level approaches the vessel height, $(\theta_1 \leq 20°, h_q \geq h_e)$. Therefore, the higher linear slosh elevation meeting this condition has to be tested to find the values of ω at different liquid volume heights. The range of A tested here went from 54 mm to 162 mm. Figure 13 shows the SIM graphs of the elevation of θ_1 versus ω when θ value is 7° at different liquid heights A. It has been found that, when $\frac{\theta_1}{\theta} = 1$, ω is within a range of values regardless of A.
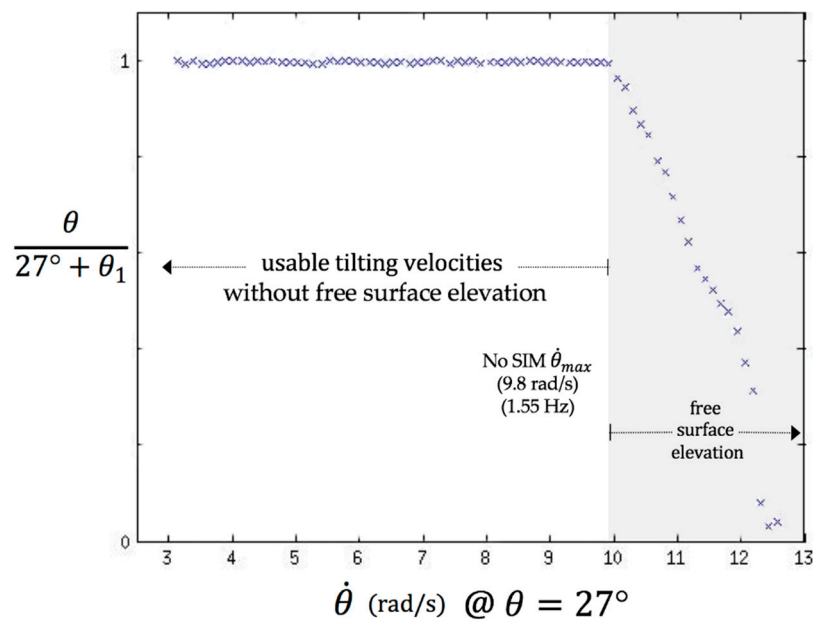
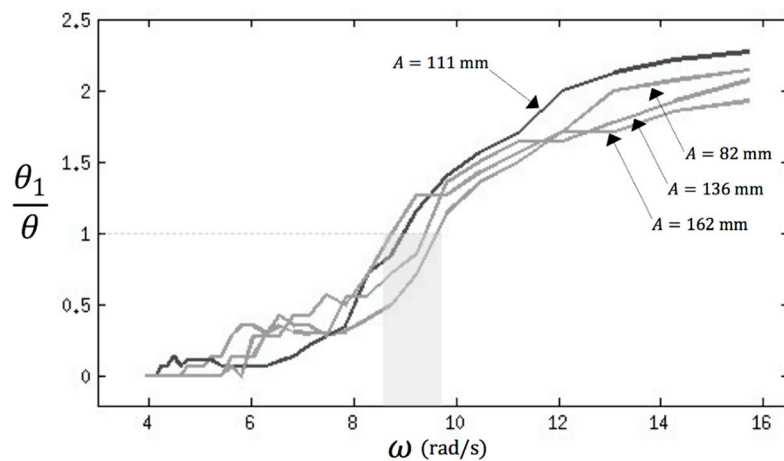**Figure 12.** Usable full cycle tilting velocities ω in the case of AFS without SIM.

Figure 13. Elevation of θ_1 versus θ along ω at different filling heights (A) when AFS uses SIM.

Figure A1 in Appendix A shows the transient response in time of a slosh in a vessel with our simulated model. Three values of θ angles, 7°, 15° and 20°, were used with their own angular velocities. The resulting free surface motion here demonstrates equivalency in symmetry, with the vessel motion close enough, as in the linear model of pendulums discussed earlier. As θ increases, ω needs to be decreased proportionally in the range of the graph of Figure 13.

Now that the timing performance of the model has been tested and proven usable, it is time to the control part of the model in AFS. As mentioned earlier, the idea is to come up with a controller that dissolves the undamped response of θ_1 and removes the steady state error. A PID controller/compensator (G_c) was designed by setting the controller constants, K_p, K_d and K_i such that the system poles in a close loop were located in the stable region with minimal oscillation. Additionally, recall that variables need to be accessible so the controller can be physically possible.

By simple inspection of product of Equations (14) and (1), $G_p \cdot G_c$, zeros and poles are added to the closed-loop system. See G_{pc} in Figure 14. As mentioned early, the location of the roots defines the stability and time response, so the controller G_c adds a dominant pole with 0.00 value in the real axis to minimize the undamped nature of the system. Two additional zeros are added by G_c so the gain is modulated to reach certain threshold.
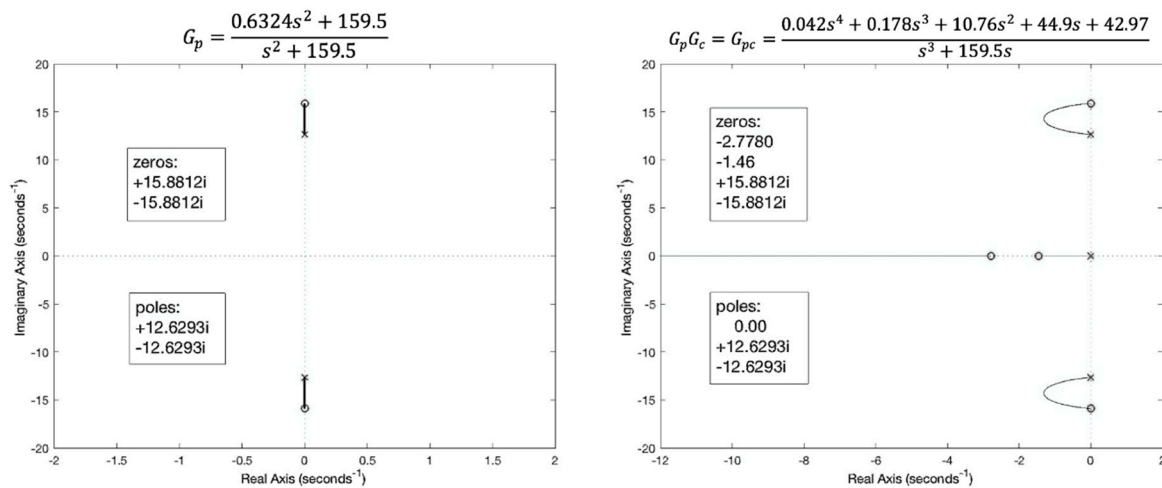
$$G_p = \frac{0.6324s^2 + 159.5}{s^2 + 159.5}$$

$$G_p G_c = G_{pc} = \frac{0.042s^4 + 0.178s^3 + 10.76s^2 + 44.9s + 42.97}{s^3 + 159.5s}$$

zeros:
+15.8812i
-15.8812i

poles:
+12.6293i
-12.6293i

zeros:
-2.7780
-1.46
+15.8812i
-15.8812i

poles:
0.00
+12.6293i
-12.6293i

Figure 14. Root locus plots of the plant gain G_p to the (**left**) and the plant-compensator gain G_{pc} to the (**right**).

In Figure 15, two step time-response graphs are shown; the one at the top shows G_p in a closed loop and its steady state error; and the one at the bottom, $G_p \cdot G_c$ is in a closed loop using the PID compensator, just as in the Figure 9 block diagram.
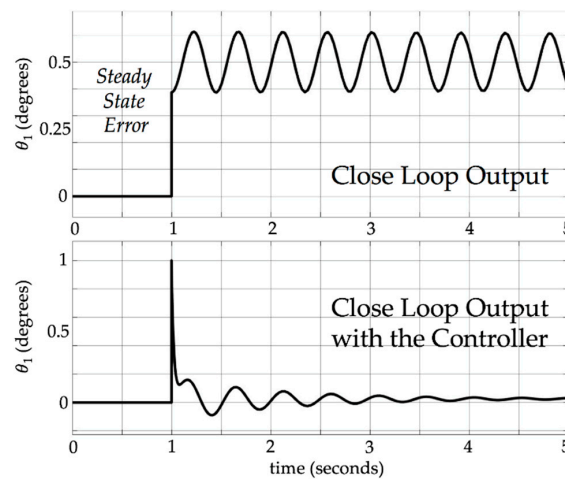

Figure 15. Step transient response of the proportional-integral-derivative (PID) controller (θ_1 versus time).

By applying the PID controller to the system, the expected results according to the proposed SIM timeline of Figure 7 were obtained. Figure 16 shows the results of the SIM timeline of θ and θ_1. As expected, θ_1 oscillates momentarily, but it settles down to a stable state as a consequence of the PID action. Figure A2 in Appendix B shows a simulated time sequence of the pouring vessel action found in Figure 16's timeline. The video S1, provided in the supplementary material Section, shows this timeline performance of the pouring action.

Considering the outcome of this research, the proposed SIM-based pouring action is a promising model to be used and implemented further. Modelling and implementing are two main objectives of this research work. The final idea of a running simulation with real time SPH flow is to collect a real-time kinematic trace for robotic manipulators in a concrete assignment of pouring; to be more accurate, it is to obtain the trace of θ after a simulation process.
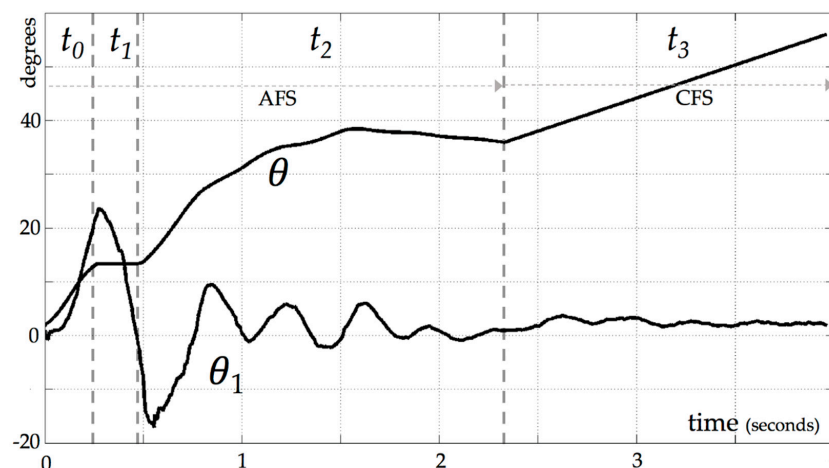
Figure 16. Timeline of pouring with SIM Control. The trace of θ to be used by an actuator.

5. Conclusions

This work validates SPH simulations for pouring kinematics. Specifically, to calibrate outflow velocities so that free fall curvatures can be estimated with enough accuracy for pouring tasks with SIM. This model is suitable for low viscosity fluids and not suitable for fluids with high viscosity where the outflow will not trace a parabolic free fall. It also allows us to measure the liquid by defining a volume or liquid height to be poured. Opposed to the indirect method of control pouring, it can measure the flow rate based on a particle matter model, and all this modelling is done within a short time using GPU processing, so motion kinematics can be implemented on-site for a robotic manipulator. Not to mention that this is a blind method where no complex vision system is required; only the filling depth and vessel dimensioning is required.

The main contribution is the slosh induction model (SIM) that proved a unique path to overcome the issue of wall-spilling during initial tilting of top-filled vessels. Extending this work to 3D modelling might be desirable but not necessary, since the linear model preserves fluid in laminar regime; the entire simulation and flow path lines never diverge to a turbulent regime.

Further simulations should explore the usage of this real-time, fluid-free surface model as a methodology to simulate 3D fluids, using different vessel shapes with different outflow tips, and a spherical pendulum model. It will also be interesting to add a no-slip condition at the fluid-solid boundaries using modern simulation schemes of SPH [31], and to impose a surface tension force on the model.

Abbreviations

SPH	smoothed particle hydrodynamics
GPU	graphic processing unit
SIM	slosh induction method
NSS	nearest neighbor search
CFS	constant flow stage
AFS	approaching flow stage
STF	system transfer function
PID	proportional, integral and derivative

Nomenclature

|			
v_q, h_q	velocity and height of the outflow	v_{0x}	v_q at vessel 1 pouring tip
h_e	empty space height in vessel 1	θ	rotation angle of vessel 1
ρ_i, m_i	density and mass of particle i	θ_1	free surface rotation angle in vessel 1
v_i, p_i	velocity and pressure of particle i	ϕ	conditional angle of θ
W	function of interpolation or SPH kernel	t_{AFS}	ending time for AFS
r_{ij}	vector distance between particle i and j	t_{CFS}	ending time for CFS
h	SPH smoothing length	L	height of vessel 1
$f_i^{pressure}$	pressure force of i	A	base length (inside width) of vessel 1
$f_i^{viscosity}$	viscosity force of i	L_0, m_0, I_0	height, mass and moment of inertia of the rigid model
g	gravity force	L_n, l_n, m_n, θ_n	height, pendulum distance, mass and angle of the *nth* pendulum
$\nabla_i W$	gradient of W	K_p, K_i, K_d	proportional, integral and derivative constants
$\nabla_i^2 W$	laplacian of W	s	laplace variable
ρ_0	rest density of water	G_p, G_c	plant gain and controller gain
c_s	speed of sound	$\mathbf{a}, \mathbf{b}, \mathbf{c}$	CFS vectors
γ	Tait's equation constant	θ'	linear step mode of θ
(x, y)	location of the outflow trajectory fall	ω	Frequency of the slosh
(x_0, y_0)	location of the outflow tip in vessel 1	t	time variable
μ	dynamic viscosity		

Appendix A
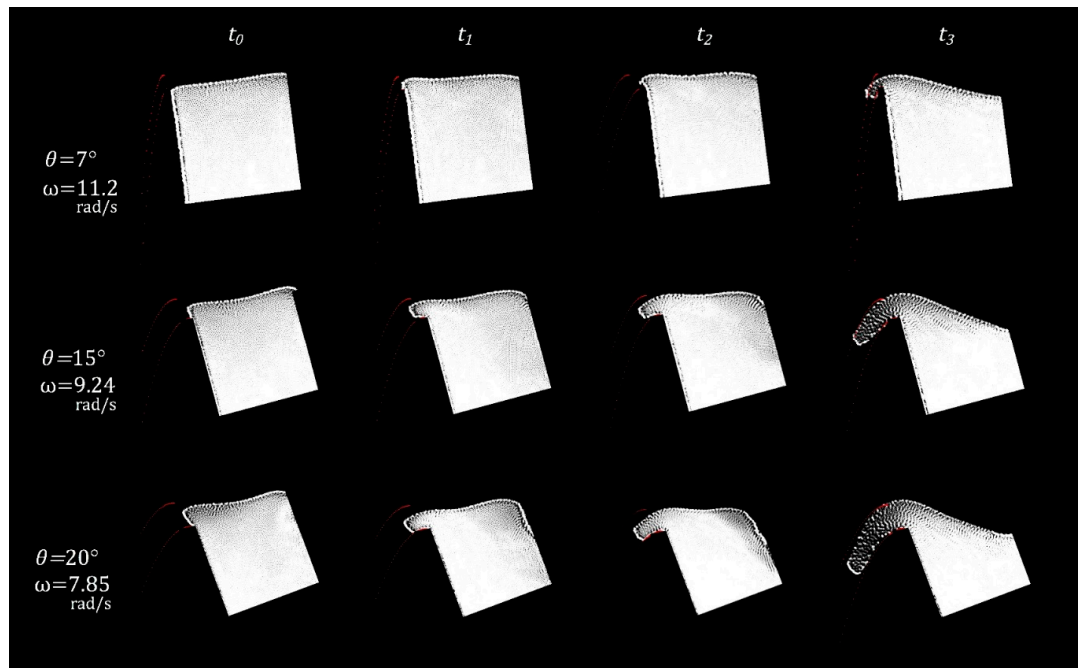
Figure A1. Transient time response of the slosh induction method using different vessel angles of rotation (7°, 15° and 20°) and at different periods of time (t_0, t_1, t_2 and t_3). t_0 time is transition 2 in Figure 4, where the slosh finds its maximum elevation in the back wall of vessel 1 due to the initial rotation. t_1 is the period of time where the slosh returns to the front wall of vessel 1 to reach, at the beginning of t_2, the desired elevation of h_q (horizontal free surface), and where AFS commutes to CFS for constant flow. t_3 shows the full elevation of the slosh on the front wall, never intended to reach due to the CFS control action. All particles are simulated within the pouring vessel 1.
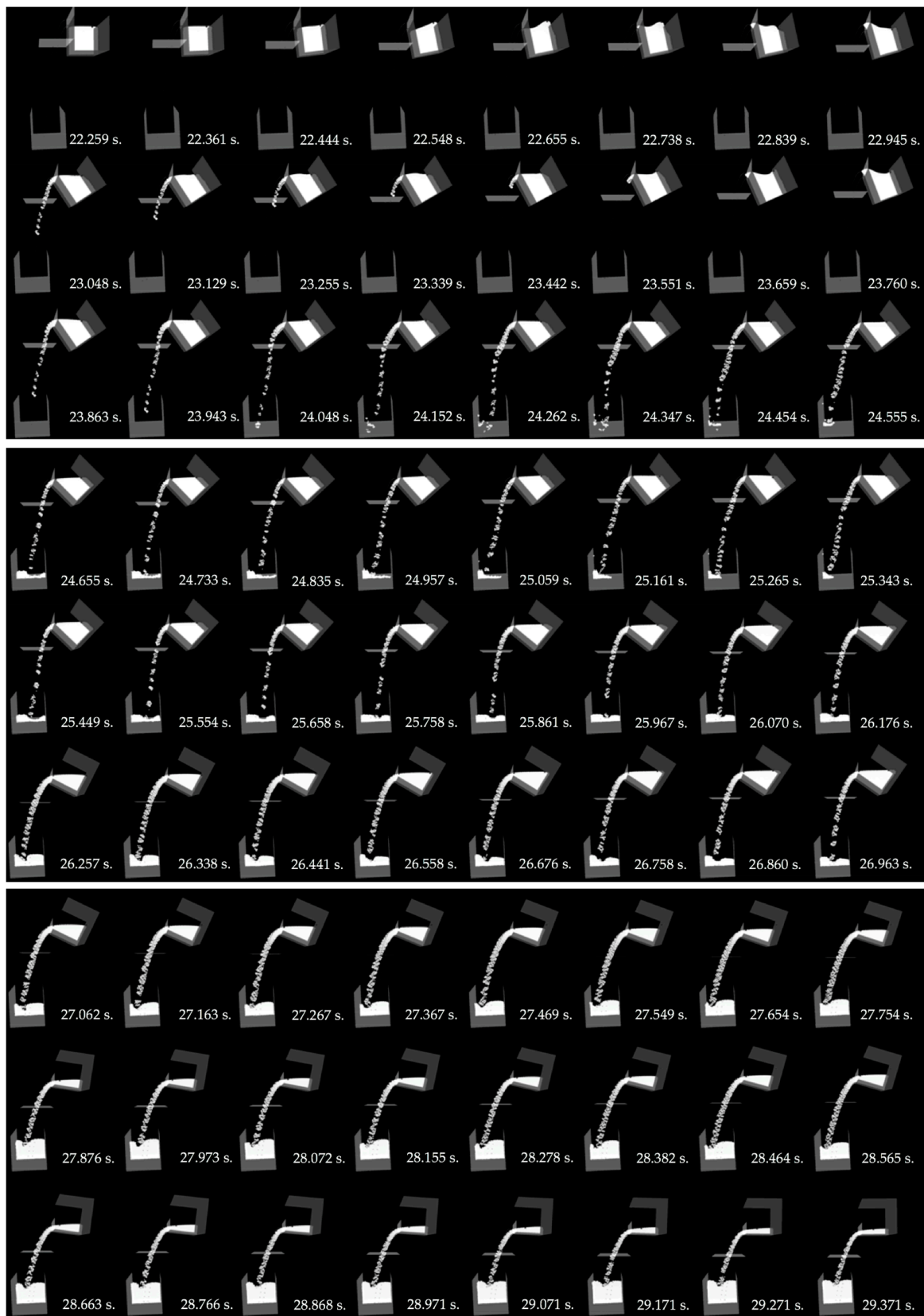
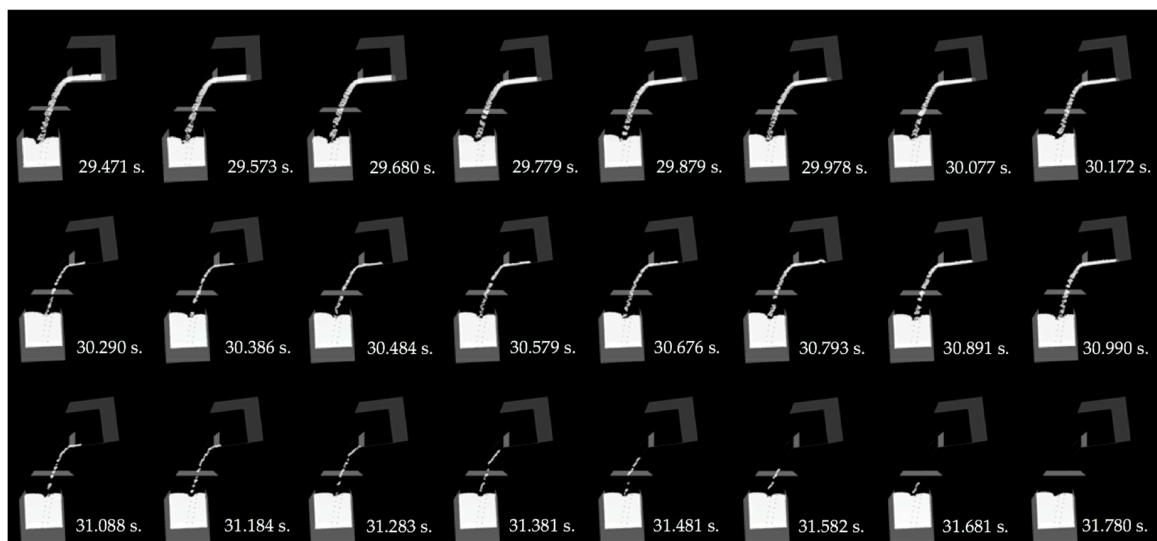**Appendix B**

Figure A2. *Cont.*

Figure A2. Simulated time sequence of the vessels using the SIM timeline (Figure 7), including all particles. A full sequence video (S1) can be found in the supplementary material Section.

References

1. Davis, E. Pouring Liquids: A Study in Commonsense Physical Reasoning. *Artif. Intell.* **2008**, *172*, 1540–1578. [CrossRef]
2. Dey, S. Free Overflow in Open Channels: State-of-the-art review. *Flow Meas. Instrum.* **2002**, *13*, 247–264. [CrossRef]
3. Bauer, S.W. The Free Overfall as a Flow Measurement Device. Master's Thesis, Lehigh University, Bethlehem, PA, USA, June 1970.
4. He, J.; Chen, X.; Wang, Z.; Cao, C.; Yan, H.; Peng, Q. Real-time adaptive fluid simulation with complex boundaries. *Visual Comput.* **2010**, *26*, 243–252. [CrossRef]
5. Hirt, C.W.; Nichols, B.D. A Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries. *J. Comput. Phys.* **1981**, *39*, 201–225. [CrossRef]
6. Becker, M.; Teschner, M. Weakly compressible SPH for free surface flows. In Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, CA, USA, 2–4 August 2007; pp. 1–8.
7. Shao, S.; Lo, E. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with free surface. *Adv. Water Resour.* **2003**, *27*, 787–800. [CrossRef]
8. Pan, Z.; Park, C.; Manocha, D. Robot Motion Planning for Pouring Liquids. In Proceedings of the 26th International Conference on Automated Planning and Scheduling, London, UK, 12–17 June 2016; pp. 518–526.
9. Lopez-Guevara, T.; Taylor, N.; Gutmann, M.; Ramamoorthy, S.; Subr, K. Adaptable Pouring: Teaching robots not to spill using fast but approximate fluid simulation. In Proceedings of the 1st Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 77–86.
10. Schenck, C.; Fox, D. Visual Closed-Loop Control for Pouring Liquids. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 29 May–3 June 2017; pp. 2629–2636.
11. Langsfeld, J.D.; Kaipa, K.N.; Gentili, R.J.; Reggia, J.A.; Gupta, S.K. Incorporating failure-to-success transitions in imitation learning for a dynamic pouring task. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Workshop on Compliant Manipulation: Challenges and Control, Chicago, IL, USA, 14–18 September 2014.
12. Ito, A.; Noda, Y.; Terashima, K. Outflow Liquid Falling Position Control by Considering Lower Ladle Position and Clash Avoidance with Mold. In Proceedings of the 3rd IFAC Workshop on Automation in the Mining, Mineral, and Metal Industries, Gifu, Japan, 10–12 September 2012; pp. 240–245.
13. Grundelius, M. Methods for Control of Liquid Slosh. Ph.D. Thesis, Lund University, Lund, Sweden, October 2001.

14. Goffin, L.; Erpicum, S.; Dewals, B.; Pirotton, M.; Archambeau, P. Validation of a SPH Model for Free Surface Flows. In Proceedings of the SimHydro: Modelling of Rapid Transitory Flows, Sophia Antipolis, Nice, France, 11–13 June 2014.

15. Krog, O.E.; Elster, A.C. Fast GPU-based Fluid Simulations Using SPH. In Proceedings of the 10th International Conference on Applied Parallel and Scientific Computing, Reykjavik, Iceland, 6–9 June 2010; pp. 98–109.

16. OpenGL Site. Available online: www.opengl.org (accessed on 23 September 2019).

17. The OpenGL Utility Toolkit GLUT Site. Available online: www.opengl.org/resources/libraries/glut/ (accessed on 23 September 2019).

18. OpenCL Community Site. Available online: www.opencl.org (accessed on 23 September 2019).

19. NVIDIA. OpenCL Sorting Networks. Available online: http://developer.download.nvidia.com/compute/cuda/4_2/rel/sdk/website/OpenCL/html/samples.html (accessed on 23 September 2019).

20. Harada, T.; Koshizuka, S.; Kawaguchi, Y. Smoothed Particle Hydrodynamics on GPUs. In Proceedings of the 5th International Conference on Computer Graphics, Petropolis, Brazil, 30 May–2 June 2007; pp. 63–70.

21. Liu, G.R.; Liu, M.B. *Smoothed Particle Hydrodynamics, A meshfree particle method*, 1st ed.; World Scientific Publishing Company: Toh Tuck, Singapore, 2003; pp. 123–124.

22. Monaghan, J. Simulating Free Surface flows with SPH. *J. Comput. Phys.* **1994**, *110*, 399–406. [CrossRef]

23. Ericson, C. *Real–Time Collision Detection*, 1st ed.; Morgan Kaufmann Publishers: San Francisco, CA, USA, 2004; pp. 175–190.

24. Cossins, P.J. Smoothed Particle Hidrodynamics (Chapter 3: Smoothed Particle Hydrodynamics). Ph.D. Thesis, University of Leicester, Leicester, UK, 2010.

25. Bell, N.; Yun, Y.; Mucha, P.J. Particle-based simulation of granular materials. In Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Los Angeles, CA, USA, 29–31 July 2005; pp. 77–86.

26. Raju, K.; Asawa, G. Viscosity and Surface Tension effects on Weir Flow. *J. Hydraul. Div.* **1977**, *103*, 1227–1231.

27. Lee, G.J. Moment of inertia of liquid in a tank. *Int. J. Nav. Archit. Ocean Eng.* **2014**, *6*, 132–150. [CrossRef]

28. Abramson, H.N. *NASA SP-106 Report: The Dynamic Behavior of Liquids in Moving Containers*; NASA: Washington, DC, USA, 1966.

29. Ibrahim, R. *Liquid Sloshing Dynamics Theory and Applications*, 1st ed.; Cambridge University Press: New York, NY, USA, 2005.

30. Roberts, J.R.; Basurto, E.R.; Chen, P.-Y. *NASA CR-406 Report: Slosh Design Handbook*; NASA: Washington, DC, USA, 1966.

31. Liu, M.; Shao, J.; Chang, J. On the treatment of solid boundary in Smoothed Particle Hydrodynamics. *Sci. China Technol. Sci.* **2012**, *55*, 244–254. [CrossRef]