Controlling a cluster of computing resources: the model free control approach

David N Donkor

CONTROLLING A CLUSTER OF COMPUTING RESOURCES: THE MODEL FREE CONTROL APPROACH

Master in Systems, Control and Information Technology (MiSCIT)

Masters Thesis

Author: David N. Donkor

Supervisors:
Bogdan ROBU (Gipsa Lab, UGA)
Eric RUTTEN (Ctrl-A, INRIA)

JUNE 2021

APPRECIATION

I would like to thank the entire Ctrl-A team for their immense support during my internship within the research team. A special thanks goes to Quentin Guilloteau (PhD) and Dr. Richard Olivier who helped me in diverse ways throughout my tenure, you share in the success of this work.

Abstract

The recent feedback control approaches adopted in cluster and file server management of High Performance Computing (HPC) systems all utilizes the model-based control approach; these model-based control laws inherit the striking flaws associated with this approach which include system modelling errors, complex parameter estimation etc., hence the need to take an alternative approach in this study. In this work, we propose to use the model free control approach with the following features: 1) a 'virtual model' representing the unknown system dynamics, 2) Elimination of the need of complex system parameter estimation and 3) assurance of less computational costs. This study first explores and adopts model free control using the intelligent PID control law, which assures both robustness and quality reference tracking in respect to time varying system dynamics. Results from both simulations and real time experiments were presented in this work to explore the intuitive abilities of this approach. From the results, the Model free control approach guaranteed good reference tracking property and robustness to external disturbances which are key performance metric in HPC applications. This MFC approach via an Intelligent Proportional controller showed a 16.51% improvement in cluster usage without overloading the file-server compared to the 15.81% improvement by the classical Proportional-Integrator controller

Contents

List of Figures

1 Introduction

1.1 Background

High Performance Computing (HPC) systems which deal with large amounts of data and computations in parallel (such as parallel computing) are facing more and more variability in their behavior related to performance and power consumption. Because they are less predictable, their administration requires a more complex management. This can be done by monitoring the information in the systems, then analyzing this data in order to activate appropriate feedback mechanisms for the whole system or only for specific applications (e.g., dismissing jobs, down-clocking CPUs) [16, 18].

In recent years, more works within the software engineering domain has developed multitude of approaches in developing self-adaptive software system [5]. Most of the proposed techniques were problem specific and not for a generic implementation. These approaches lacked the theoretical grounding to ensure their dependability beyond their microscopic applications. With the mathematically grounded techniques that control theory assures for adaptation, they guarantee their efficiency and operability under precise operating conditions.

There exists a striking similarity between control theory and autonomic computing as seen in [8] for system adaption, the former associates an adaptive system within the context of a closed loop, and the latter creates by coupling the system with its adaptation manager, with proof of concept in reference to the design framework of the MAPE-K feedback loop[15]

The recent feedback control approaches adopted in cluster resource management of High Performance Computing (HPC) systems all utilize the model-based control approach; these model-based control laws inherit the striking flaws including system modelling errors, complex system parameter estimation, high computational costs.

A pragmatic approach in utilizing data-driven techniques that satisfy the assurances that control theory provides such as stability, accuracy, and reliability in [8], has motivated the adoption of control theory in the design process of self-adaptive systems, particularly for this work. To this effect, this research work will investigate and explore the assurances of control theory approach utilizing a data-driven technique in the management of cluster resources in High Performance Computing(HPC) systems.

1.2 State of the Art

1.2.1 Cluster Autonomic Management

A Control theory approach was used in [20] for HPC scientific workflow management. The objective of this feedback control was to ensure the completion of low priory scientific job in a most efficient way. The model is first developed to describe the parameters that relates to the key aspect of the infrastructure. The given system was first assumed as a first order plant, and with subsequent modification to the systems dynamics, it was assumed to be a second order plant, taking into consideration the file-server profile. In this work, a Proportional-Integral control law was designed and was later improved to utilise the Model Predictive Control law (MPC) to adopt to the system constraints and system's parameter variations.

1.2.2 A Model Free Control

The concept of control theory that utilises process data from a given system in order to meet performance goals without in depth knowledge of that given system via an intelligent Proportional-Integral-Derivative controllers was fully developed in [10]. The concept is based on functional analysis and differential algebra. In subsequent works under this concept, a novel fast parameter estimation technique was proposed in estimating the unknown system dynamics. A virtual model is proposed to represent the system with its loop closed via an intelligent controller. A connection of this intelligent controller and that of the classical Proportional-Integral-Derivative controllers was made in this work. This Model free concept has been explored in various applications such delay systems, infinite-dimensional systems, etc.

[1], adopted the recent model-free control technique to deal with under-actuated mechanical systems for stable limit cycles generation. In order to control this highly non-linear system, a model-free controller is first designed in order to track some parametric reference trajectories. This work then went on to develop a second model-free controller whose key design strategy is based on using trajectories' parameters as control inputs to ensure that the internal dynamics of the mechanical system is well stabilized. This proposed method was applied to a real under-actuated mechanical system: the inertia wheel inverted pendulum. Numerical simulations as well as real-time experiments were presented in this work to validate the effectiveness of the model free control method and its robustness toward external disturbances. From both simulation and experimental results, the Model free control

approach ensured a good reference tracking of the parametric reference trajectories and was also robust to external disturbances.

Cloud computing management, their dynamic adaptation of computing resource allocations under time varying workload is an active domain of investigation. Several control strategies has already been proposed. This study [2], adopted the model free control setting and the corresponding "intelligent" controllers, seen in many concrete engineering applications , for horizontal elasticity . Horizontal elasticity is a scalability approach in cluster resource management which consists of adding or removing instances of computing resources associated to an application. The Model free control approach was then compared to the commercial "Auto-Scaling" algorithms, and from the results, the easily implementable Model free control approach behaved better even with sharp workload fluctuations. This observation was confirmed by experiments on Amazon Web Services (AWS) subsequently.

1.3 Scientific contribution

In the recent works carried out on cluster resource management of High Performance Computing (HPC) systems, utilized the model-based control approach. Models were proposed to represent the dynamics of the file-server and the cluster system. These model-based control laws inherit the design restraints such as system modelling errors or complexities, complex system parameter estimations, high computational costs etc. The underlining disadvantages of the model-based scheme and their real time implementation complications were noted in [20], [18], hence the need to take an alternative approach in this work.

In this work, we propose to use the model free control approach with the following features:

1. A 'virtual model' which will be a representative model for the clustering system.
2. A fast and simple parameter estimation technique to estimate the unknown functional cluster dynamics
3. An easy and less computational costs for real time implementation .

This study first explores and adopts model free control using the intelligent Proportional-Integral-Derivative control law first in simulations to validate its assurances such as robustness and quality reference tracking. Based on the simulation results, the Model free control

approach was implemented on the real time system (Grid'5000 Platform) to ascertain and explore the intuitive abilities of this approach in the cluster resource management.

2 System Description

2.1 System Architecture

2.1.1 The GRICAD Center

The GRICAD center located in Grenoble is one of the most powerful High Performance Computing (HPC) tier-2 centers in France. It is composed of over 6500 cores that add up to 135 TFlop/s and 28 TB of memory distributed over 12 clusters. It provides HPC resources to academic research communities from a wide range of disciplines: Biology and Health, Chemistry, Environment and Climate, Numerical Physics, Earth and Planetary Sciences, and Distributed Computing [3].

Since the year 2000, GRICAD known previously as CIMENT has allowed all users of parametric tasks to launch their jobs on idle processors across all the computing clusters of the grid. These services are launched via specific middleware called CiGri and a scheduler called OAR to deal with the special type of jobs called the "best-effort jobs". The notion of "best-effort jobs" means that, there is no Quality of service (QoS) objective. These "best-effort jobs" which are low priority jobs are submitted on the cluster to utilize the idle resources on the cluster. It should be noted however that, anytime there is the need of these resources for the high priority jobs issued by the local user, their execution is killed and resubmitted again when there are idle resources.

2.1.2 CiGRi Middleware

CiGri [6] is a lightweight grid middle-ware designed to run on top of a set of OAR clusters to manage efficiently large sets of multi-parametric tasks (also called bag-of-tasks). CiGri is written in Ruby by the MESCAL team from the LIG laboratory and INRIA. With a powerful events handling, it manages automatic re-submission that is useful for best-effort jobs. This software offers a simple interface to users that have to launch big set of parametric tasks.

The utilisation Policy[3] is summarised in figure (1).

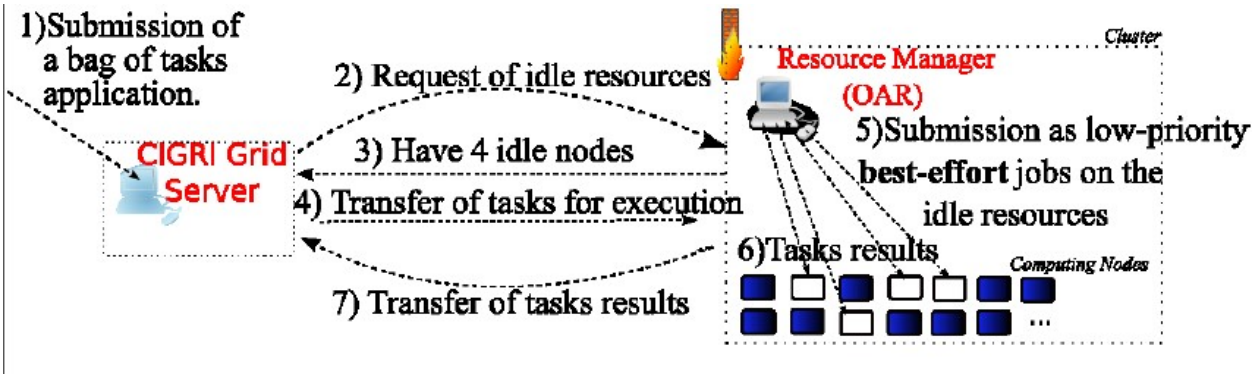


Fig. 1. A summary of CiGri Utilisation Policy [3]

2.1.3 OAR Scheduler

OAR is a versatile resource and job management system (also known as a batch scheduler) for HPC clusters and for testbeds where versatility is an important feature. OAR architecture is based on a database (MySQL), a script language (Perl) and with an optional scalable administrative tool (e.g. Taktuk). It is composed of modules which interact mainly via the database and are mostly executed as independent programs. Formally, it uses no API, since the system interaction was completely defined by the database schema. This approach eases the development of specific modules, hence each module (such as schedulers) could be developed in any language having a database access library [14]. In our case, local users submit jobs directly onto the cluster via this scheduler.

2.2 Experimental Platform

The Grid'5000

During this project, the experimental platform used was the Grid'5000 platform; we used four machines on the Grisou cluster located in Nancy [17], France. several real time experiments were performed in order to identify and validate the proposed model free control approach. The experimental set-up environment was composed of a CiGri server, an OAR scheduler, a cluster server emulating 100 nodes and to consider the storage profile, an additional machine was deployed to represent the file-server. The workflows used for experimentation were composed of dummy jobs (sleep Linux command) with a duration of thirty (30) seconds. The impact on the file server load was simulated with a linux dd command (/dev/zero) as

input. A file of size 100Mb is written to the file-server within each workflow. In some cases, we stressed the system with external disturbances by submitting higher priority jobs to the cluster, in order to dynamically vary the amount of available resources and file-server load's to check the robustness of the proposed control approach.

2.3 Problem Formulation

The main objective of this project is to maximize the exploitation of idle resources to improve the cluster utilization through CiGri, while ensuring the file-server is not overloaded. In summary, the goal of this research is to address:

- **Cluster Resource Under-utilization:** The workflow of our given system utilises an algorithm that cyclically submits the number of jobs rate to the cluster, which increases at every iteration. Note that, in each cycle, the algorithm waits for the completion of all submitted jobs, but under several fail safe mechanisms intended to protect the infrastructure[3]. Given this workflow profile, jobs are submitted onto a cluster only when its waiting queue is fully empty. In essence, this behavior leads to scenarios where the cluster is being under-used (or not used at all) in spite of the existence of remaining jobs in the bag-of-tags. This is a cluster resource management problem; we aim to improve in this work.
- **The File-server Overload** The use of grid resource computing involves computations and data processing, hence the need of storage resources, whether for both reading of input and writing of outputs from a server, or for storing the output of the script. The effects of I/O process undoubtedly impact the file-server's performance when running hundreds of simultaneous jobs. This phenomenon makes storage a major challenge in parallel computing infrastructures, and a limiting factor in the performance of the grid resource management

2.4 Drawbacks from recent works

From previous works [20], [18], many model-based control theory approaches were proposed to address the workflow of the cluster resource management performance metrics. Various

system models representing both the cluster and file-server dynamics were proposed, which resulted in complex parameter estimations and assumptions which are experiments specific.

Also, the proposed control algorithms in previous works such as the linear quadratic tracking(LQT) lacked the necessary libraries to implement it on the real time platform, and also in some cases [20] necessitated high computational costs.

These drawbacks, informed us on the path of choosing a control theory approach which will eliminate these design process constraints and hence the model free control technique presented in the next section.

3 Our Approach

3.1 Introduction

Model-free control strategies proposed in [13], [4], resulted in a breakthrough of nonlinear systems control. This technique is based on replacing the unknown complex mathematical model with a simplified local model which relies on a fast estimation and identification of the non-linearities of the unknown system [12]. Generally, the term Model free highlights on the concepts that, there is no need of priori knowledge of the system dynamics in a given context.

The control scheme is primarily based on a local linear approximation of the system dynamics which is validated for small time window. The approximation is updated in an online fashion with the help of a fast estimator [13].

The main objective of this control strategy is to proposed a technique which does not depend or require prior knowledge of the system dynamics, nor complex system parameters tuning. In this way, it gives the flexibility to build a controller for any unknown system. This has given the Model free control approach recognition as an intelligent and fast control techniques for variant nonlinear systems dynamics.

Model free control technique has been utilized in a number of practical case studies such as humanoid locomotion[19], under-actuated mechanical systems[1], delay systems[9], financial forecasting[9] respectively.

3.2 The Model Free control description

Consider the input-output behavior of a system, which for simplicity's is assumed to be mono-variable, and can be governed within its operating range by an unknown finite-dimensional ordinary differential equation[13]. This system characterised with one control input variable (u) and single output variable (y), replaces any given system's complex mathematical model with what we primarily referred to as the **ultra-local model** expressed in the form:

$$Y^{(v)} = F + \alpha u \tag{1}$$

Where,

1. $Y^{(v)}$ is the derivative of order $v \geq 1$ of y. The integer (v) is based on the practitioner preference. However, in practice the value is chosen to be low thus 1 and in less cases is chosen as 2.

2. α is a non-physical parameter chosen by the practitioner, which is a constant value. It is chosen by the practitioner such that αu and $Y^{(v)}$ are of the same magnitude. This α parameter is obtained normally through experimental trial and errors and not priori defined, hence there is the need to understand the systems behavior possibly through collaboration with the system's engineers

3. The F parameter, which is continuously updated, subsumes the poorly known/unknown parts of the plant as well as of the various possible disturbances that are subjected on the system, without the need to make any distinction between them. For its estimation, the F is approximated by a piece-wise constant function which will be further discussed in the following section

Remark 1: In our case of the High performance computing system, there are a lot of system parameters which cannot be known priori as well as external disturbances such as the local users. With this in mind, we also considered our system to be an order of 1 for simplicity, in order to explore the intuitive ultra-local model concept which is 'virtual' represented as seen in equation (1) within the context of cluster resource management.

3.2.1 Unknown dynamics Online Estimation

In the first publications on model-free control, the technique proposed for the estimation of **F** representing the unknown dynamics of the system was based on the numerical differentiation of noisy signals[12]. In this work, we adopted the parameter identification utilization techniques discussed in [10]. The algebraic identification techniques proposed in [12], [11] is applied to the equation:

$$Y^{(v)} = \phi + \alpha u \tag{2}$$

where ϕ is an unknown constant parameter, and its estimation:

- necessitates only a short time lapse
- is expressed via algebraic formulae which contain low-pass filters like iterated time integrals

- is robust with respect to noise

This constraint ensures the smooth estimation of the ϕ parameter in equation (2), which is applicable in estimating the F parameter which represents the unknown system dynamics.

Remark 2: In this work, we considered the estimation of the F parameter in the discrete time domain in regards to the dynamics of the cluster management system. The Estimation of **F** in relation to computational applications is achieved by the expression:

$$F(t) = y^{(a)}(t) - \alpha u(t-h) \tag{3}$$

where a indicates the order of the system in which for our case we considered the system to be of order of 1, and with h representing the sampling interval for each iteration.

3.2.2 MFC Control laws

From the perspective of modern control theory, controller design is based on the mathematical model of a plant. Typical linear control system design methodologies are zero-pole assignment, linear quadratic regulator (LQR) design, optimal control design, and so on. For nonlinear systems, the inevitable controller design methods are the Lyapunov-based methods with backstepping controller design, feedback linearization, etc. These controller design methodologies are recognized to be typical Model Based Control (MBC) system design methods. The key feature of Model Based Control (MBC) controller design and analysis is that, the control system design is based on the modeling accuracy and correctness of the assumptions imposed on the plant mathematical model which determines the control system performance, reliability, and safety, since the available system model kinetics is embedded in the control system. If the system model is unavailable or the assumptions do not hold, then no conclusion on controller design and analysis could be obtained, and subsequently there exist no way to discuss their applications. The model Based control method, hence comes from the system model and ends up in the system model.

A more pragmatic approach is now adopted in utilising the available process data in the context of control, thus the Data-driven control methods. This approach is the control theory and method, in which the controller is directly designed by using online or offline I/O data of the controlled system or knowledge from the data processing without explicitly using the information from the mathematical model of the controlled process, and whose

stability, convergence, and robustness can be guaranteed by rigorous analysis under certain reasonable assumptions.

Until now, there exist only a few Data-Driven Control methods, such as proportion integral differential (PID) control, model-free adaptive control (MFAC), iterative feedback tuning (IFT), virtual reference feedback tuning (VRFT), iterative learning control (ILC) etc.; This method lies in the fact that the dependence on the mathematical model of a controlled plant is removed for control system design hence best suited in cases where the accurate model is unavailable, typically in our case of a High performance computing(HPC) cluster resource management.

In reference to the Model free control (MFC) control laws approach adopted in this work, it obvious to admit that this control method falls under the data-driven control approach. Our ultra-local model which is virtual, discussed in the previous sections is augmented with a control law termed as the intelligent proportion-integral-differential (PID) controller. This approach uses this corrector to control our unknown model.

Now given our ultra-local model in equation (1), let's assume $v = 2$ which yields:

$$\ddot{y} = F + \alpha u \tag{4}$$

We can then close the loop using an intelligent proportional-integral-derivative controller given as:

$$u = -\left(\frac{F - \ddot{y^*} + K_p e + K_i \int e + K_d \dot{e}}{\alpha}\right) \tag{5}$$

where

- y^* is the reference trajectory
- e = y - y^* is the tracking error
- K_P, K_I, K_D are the usual control tuning gains

We can deduce from equation (4) and equation (5) upon substitutions and simplification that, we can rewrite our equation in the form:

$$\ddot{e} + K_d \dot{e} + K_p \dot{e} + K_I \int e = 0 \tag{6}$$

From equation (6) we can note that F parameter does not appear anymore in the equation; which means that the unknown system dynamics and disturbances subjected on the system is eliminated. The remaining thing left, is the tuning of the control gains K_P, K_I, K_D which can achieved by several control theory techniques to obtain a good reference (y^*) tracking.

Remark 3: The intelligent control law inherently captures the non-linear dynamics and disturbance of a given system, and then successfully eliminate these performance constraints. This control design implements a double corrective action, through its online estimation of unknown dynamics and also complementing that with a Proportional-Integral-Derivative term which makes it more robust control technique. An abstract of discrete implementation of the intelligent PID Algorithm is shown in figure (2). (See Appendix for the full implementation).

It is however to be noted that, in practice most applications utilises the intelligent P which ensures satisfactory reference tracking and robustness, and is analogous to the classical PI controller. In [7] there exists a relationship between the gains of the classical Proportional-Integral-Derivative controllers and that of the intelligent PID controllers, which was noticeably dependent on the sampling interval [h] and the α parameter respectively. A validation experiment will be simulated on these concepts in the next section.

```
% Computing the Errors in both PID and MFC cases
 Error(i+1) = Desired(i) - Sys_PID_Output(i)   ;    % Error entering the PID controller
 MFC_Error(i+1) = Desired(i) - Sys_MFC_Output(i);     % error entering the MFC controller

 CumError=CumError + Error(i+1);
 MFC_CumError=MFC_CumError + MFC_Error(i+1);

 %% Building the Classical PID
 u_PID(i+1)=kp*Error(i+1) + ki*CumError*dt + kd.*(Error(i+1)-Error(i))/dt;  % PID control law in dicrete time

 %% Building the MFC controller

 u_MFC(i+1)=kd/(Alpha*dt)*MFC_Error(i+1) + kp/(Alpha*dt)*MFC_CumError*dt + (-1)/(Alpha*dt)*(MFC_Error(i+1)-MFC_Error(i))/dt;    %% equation 28 from MFC Fliess and Join

 %% Building the Plant model with a PID
 Sys_PID_Output(i+1)= b1*u_PID(i+1) + b2*u_PID(i) + a*Sys_PID_Output(i);  % The Plant response when agumented with a PID

 %% Building the Plant model with an MFC controller
 Sys_MFC_Output(i+1)=b1*u_MFC(i+1) + b2*u_MFC(i) + a*Sys_MFC_Output(i);    % The Plant response when agumented with MFC
```

Fig. 2. An Abstract of the discrete iPID approach

3.2.3 Prospects and Drawbacks of MFC

The prevalent constraints on modelling the time varying dynamics of nonlinear systems in control theory has been of one the challenging stages in the design process. This model free control approach has not only been acknowledged in the scientific domains but its implementation is seen in industrial applications leading to patents in some cases. The

following are the major assurances of the model free approach:

- No need of a priori knowledge of the system dynamics
- No complex parameter identification is needed
- Quality reference tracking
- Robustness in regards to disturbances

However, though the Model free control has been recognized as an intelligent and fast control technique for variant nonlinear systems dynamics, it comes along with a drawback in terms of its dependency on the quality of the sensor used in accessing process data and also the sampling frequency which relies on the fast local estimation

3.3 The MFC approach in the context of cluster resource management

As emphasised in this work, we adopted the control theory approach discussed in depth in the previous section. We first developed this concept both in MATLAB Simulink software and in script, in order to validate the assurances this approach proposes in simulation (See Appendix for the Simulink blocks and the script). As a data-driven approach our first task was to explicitly identify the Inputs/Outputs of the cluster resource system under this study. A schematic of the work flow for this study is shown in figure (3) below.

Model and Control
At this design stage, we used the ultra-local model which is virtual in equation (1) to represent our system's dynamics. In this regard, no mathematical representation for the cluster and file-server systems were needed in designing our control law. The control law utilised as discussed in section three, specifically in equation (5) was designed to meet our system objectives, thus:

- A set-point based goal for our file-server translated as a reference tracking objective in the control theory context
- To improve the cluster resource usage

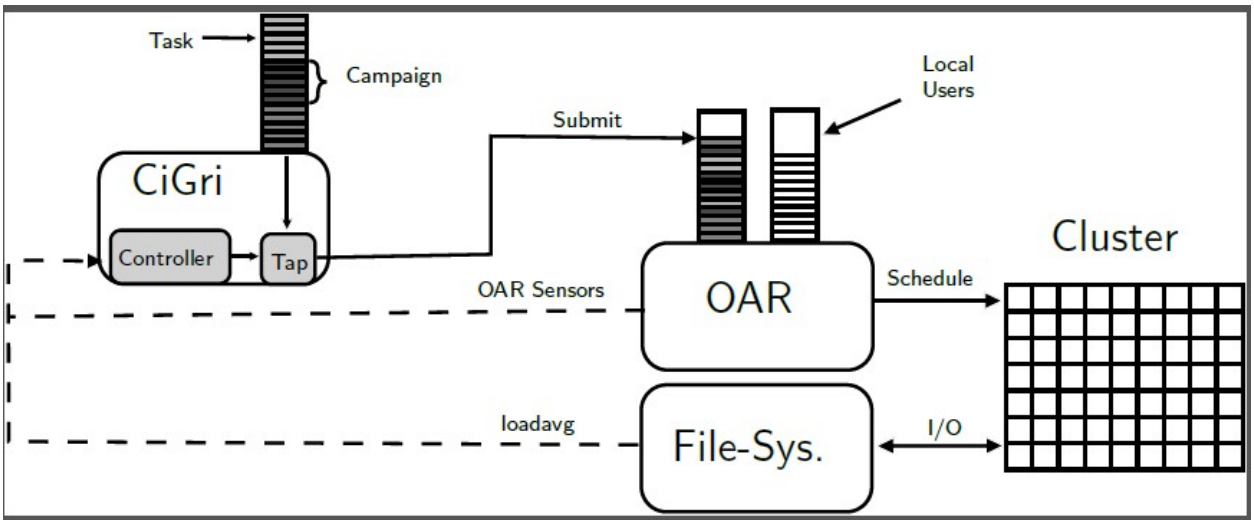


Fig. 3. A schematic overview of our cluster workflow

Defining I/O of our systems

As depicted in figure (3), the system is designed to take data inputs from the OAR sensor and the file-server sensor. The control algorithm then integrates these two data profiles and outputs a response which is translated in the number of jobs to be submitted to OAR scheduler in order to achieve the two key performance objectives/goals for the system. In the next section, we will present the results obtained in integrating the Model free control technique both in simulation and real time.

4 Results And Analysis

In this section, we first validate the assurances the model free control technique provides in simulations and subsequently on the real time system. The key part of the section was to first explore this MFC concepts with different plant models to ascertain its performance and stability before configuring it to suit our given system.

4.1 Simulation Results

In order to validate this control theory approach, we were first interested in the relationships between the Classical and Intelligent PID control laws based on the gains. The aim was to find an analytical way to tune the control gains when one control scheme gains is found.

Later in this section, we will present simulation results on a first and second order plant to validate the robustness, stability and reference quality properties of this approach.

4.1.1 Validation of the relationship between classical and Intelligent PID

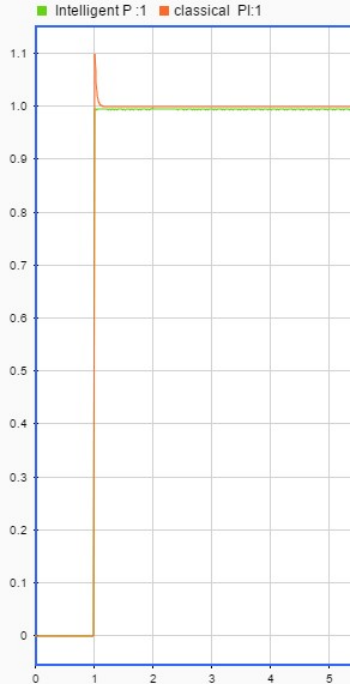
In this section, I first run simulations using the MATLAB Simulink model I developed to validate the relationship based on the gains between the classical PID and the intelligent PID in [12], according to the ratio of the product of the sampling time and the α parameter. According to this work [12], there exists a relationship between the classical PI and intelligent P, Classical PID and intelligent PD, and a Classical PI^2D and intelligent PID. For this section, I will present the simulation results on the first two cases using a first order system.

To run this validating experiment, the following tuning parameters we chosen:

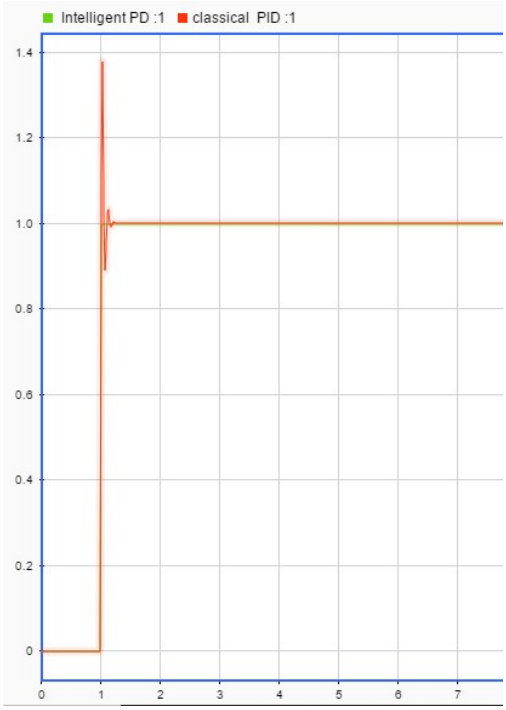
- For the PI and iP simulation, $K_p = 35$, sampling (delay gain) time= 0.05 and $\alpha = 0.10$. The relational gains were computed respectively for the classical PI using equation (7). The simulation result is shown in figure (4a) **Remark 4:** The ratios of the relational gains are given in equation (7):

$$kp = \frac{1}{\alpha h}, ki = \frac{K_p}{\alpha h} \tag{7}$$

- For the PID and iPD simulation; $K_p = 50$, $K_D = 0.05$, sampling (delay gain) time=0.01



(a) Simulated results on the relationship between Classical PI and the intelligent P based on the relational gains



(b) Simulated results on the relationship between Classical PI and the intelligent PD based on the relational gains

Fig. 4. The connection between the intelligent PID and the Classical PID

and $\alpha = 0.10$. The relational gains were computed respectively for the classical PID using equation (8). The result is shown in figure (4b).

Remark 5: The ratios of the relational gains are given in equation (8):

$$kp = \frac{K_D}{\alpha h}, ki = \frac{K_p}{\alpha h}, kd = \frac{1}{\alpha h} \tag{8}$$

From figure (4a) and figure (4b), we can notice that, each controller pairs implemented both had similar response time and the settling time, and also with a good reference tracking. This confirmed the concept of equivalence as proposed in [12].

4.1.2 MFC simulation on different Plant systems

The MFC concept designed in MATLAB Simulink was simulated on a first and second order plants. The choice of the plant order was based on the system order's used in the previous works in relation to this system. The gains of the controllers were tuned using the MATLAB in-built PID gain tuning interface.

Remark 6:
We used this model in the simulation process solely to generate the Input/Output data. The controller design used was not based on this simulation's mathematical models.

The Plant's Simulations
Here, we will test the MFC both on a first and second order plants, first without disturbance and then a disturbance will be introduced at t=20s. For this comparative simulations, we used the same control gains.

From the simulation results for first order plant with and without disturbance shown in figure (5) and figure (6), we can deduce from the results that,

- The intelligent PI and intelligent PID showed a faster response time compared to their Classical PI and PID controllers
- When a disturbance in the form of a step input introduced at t=20s, the intelligent controllers showed a good disturbance rejection with respect to the magnitude of deviation from the reference value. Also, the intelligent P showed good response characteristics with its settling time after the perturbations.

Let's consider the simulation results for Second order plant with and without disturbance shown in figure (7) and figure (8), we can observe from the results that,

- The intelligent PI and intelligent PID showed a faster response time compared to their Classical PI and PID controllers
- When a disturbance in the form of a step input introduced at t=20s, the intelligent controllers showed a good disturbance rejection with respect to the magnitude in deviation from the reference value. Also, we can notice that intelligent controllers have good response characteristics with regard to their settling time after the perturbations.

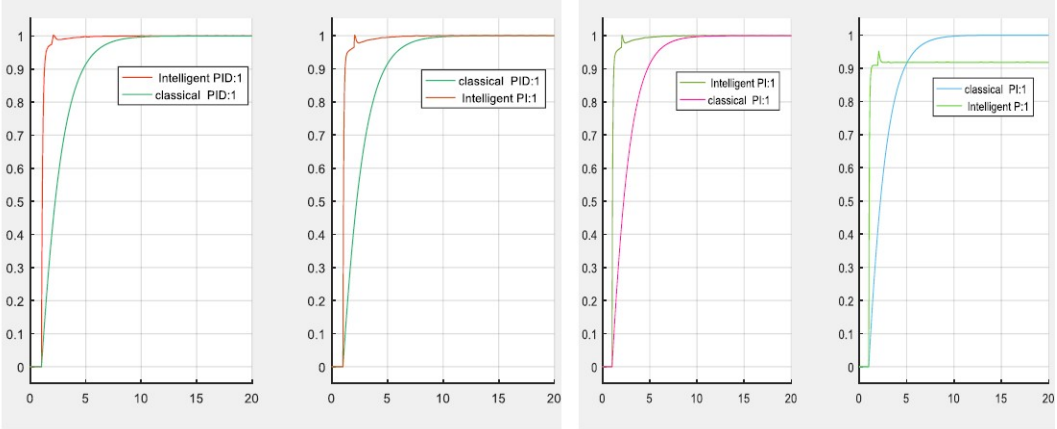


Fig. 5. Comparative Simulation results on a first order Plant without disturbance

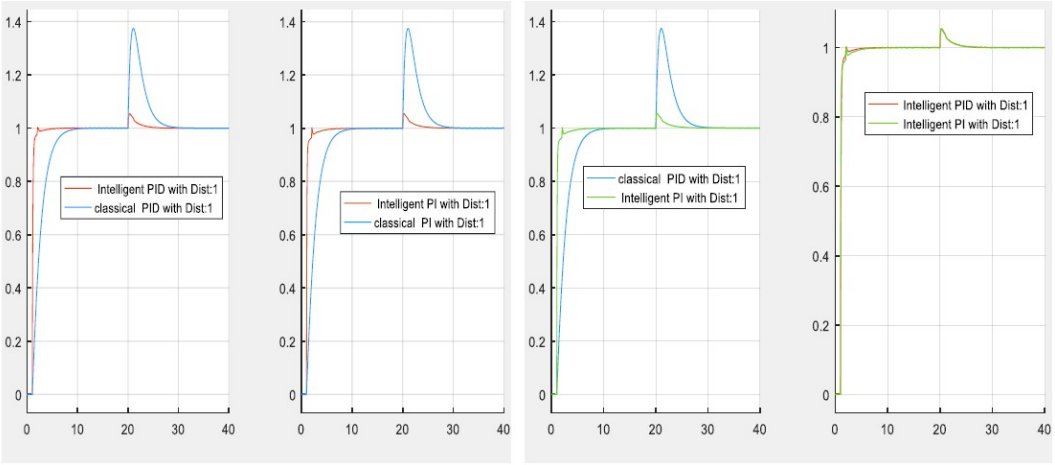


Fig. 6. Comparative Simulation results on a first order Plant with disturbance

- From figure (8), the intelligent PI showed an impressive response time and disturbance rejection characteristics with regards to the classical PID

4.2 Real time Experiments results

Experiments were run on the Grid'5000 platform using the Grisou cluster located in Nancy. We used four machines to simulate the Fileserver, OAR server, CiGri and lastly, a machine to emulate the cluster with 100 nodes representing our cluster resource. The workflow for the experiments is characterised in two design stages; First, sleep for 30s and then submit a file size of 100Mb to the fileserver. Here α was computed as $1/r_{max}$ where r_{max} is the

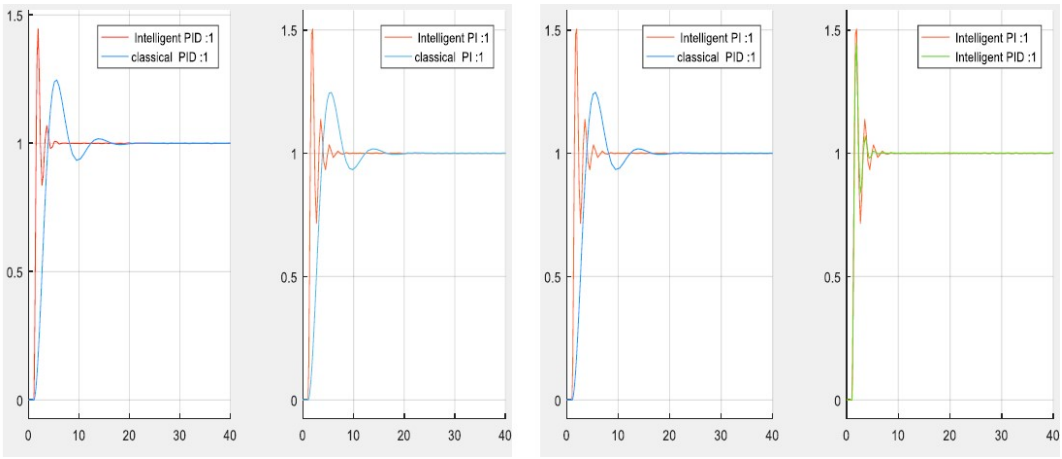


Fig. 7. Comparative Simulation results on a Second order Plant without disturbance

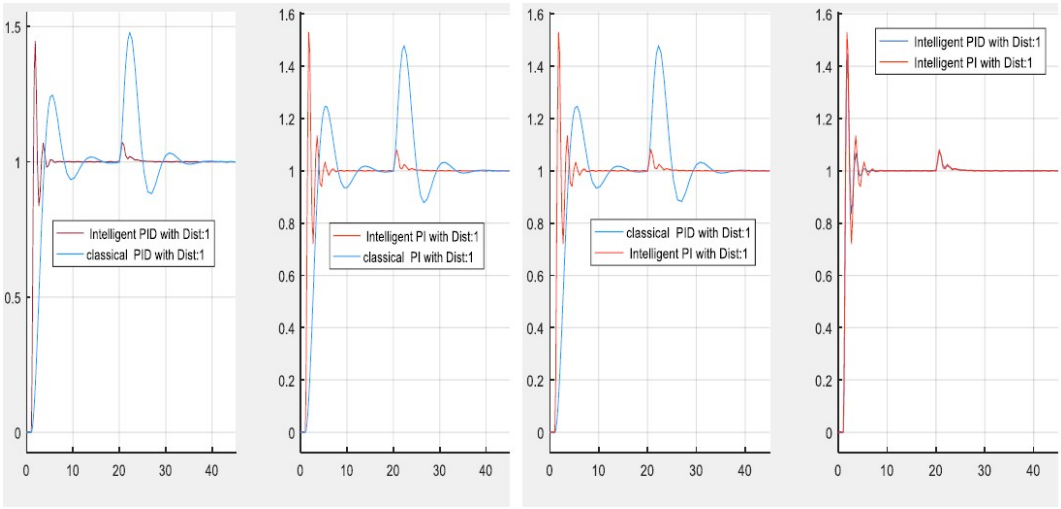


Fig. 8. Comparative Simulation results on a Second order Plant with disturbance

total amount of resources available, which can be translated into the maximum number of jobs the scheduler can allocate depending on the availability of resources. This was done to normalise the load within a given set-point for the control algorithm. In order to evaluate our approach, we utilised two performance metrics:

- **Fileserver load tracking**: Overloading in the context of computer science occurs when the loadavg which is a process data measured using the fileserver sensor (see figure (3), is higher than the number of cores available on the cluster system. So here, we are interested in tracking this load in order to avoid overloading the fileserver.
- **Cluster Utilisation** : The scheduler does not allocate jobs immediately after a resource becomes idle. The process involves scanning for idle resources, verifying that resources' specifications matches the jobs requirements, and before allocating the jobs.

In practice, this takes non-negligible time, during which the cluster has unexploited idle resources. For this reason, even under the best management strategy, the overall cluster utilization during a campaign cannot be 100%. The cluster usage can be computed analytically by:

$$ClusterUsage(\%) = \frac{1}{t_0 - t_f} \sum_{t_0}^{t_f} \frac{r_k \Delta t}{r_{max}} \tag{9}$$

where r_k is the number of running jobs in the cluster at a time k , t_0 and t_f being the initial and final time, Δt is the sampling time and r_{max} indicating the maximum number of running jobs allowed

For this work, we shall present the case of the intelligent Proportional controller against the classical Proportional controller in this section, and then examines the effect of the intelligent Proportional controller and the classical Proportional-Integral controller in relation to their cluster usage.

4.2.1 Simulation on Grid'5000 with and without Local Users

Let us first consider the implementation of the MFC approach via an iP and a classical P. Here we choose a gain $(K_p) = 0.1$ with a sampling time of 30 seconds. We simulated the presence of local users by submitting higher priority jobs to the cluster. The results in both cases is presented in figure (9) and figure (10)

In figure (9), we can observe that the intelligent P managed to track the load reference value of 3, throughout the workflow duration. In relation to the classical P, at time 200 seconds, we observe an overshoot in its trajectory which can be translated as a case of file-server overload.

In figure (10), we considered a case where local users were present at time 950 seconds. This can be observed in the rise of the number of running jobs and its effect on the load trajectory in both the intelligent and classical P approach. However, from the load profile in figure (10), the intelligent P recovers smoothly and managed to track its interval reference set-point over the remaining workflow duration. In the classical P case, we observed another overshoot at time 1600 seconds, indicating non-robust characteristics in the presence of external disturbance (local Users).

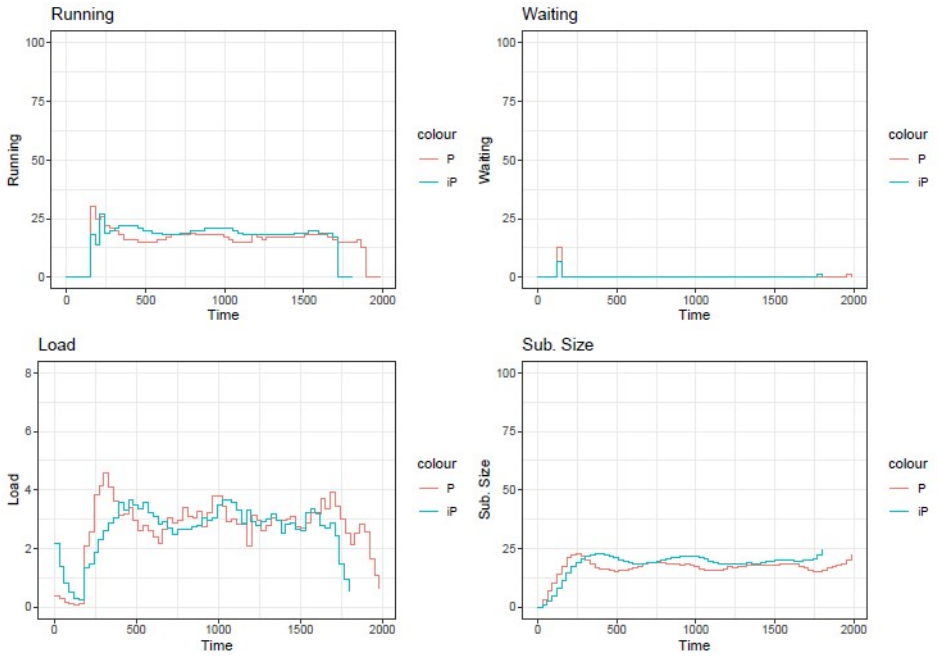


Fig. 9. Comparative Simulation results on the Grid'5000 between an Intelligent P and a classical P without local users

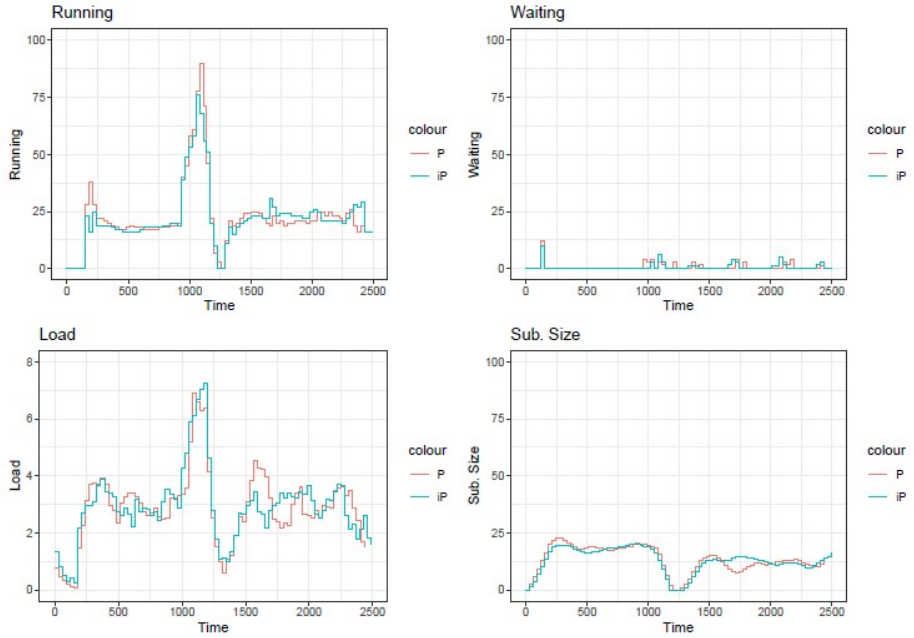


Fig. 10. Comparative Simulation results on the Grid'5000 between an Intelligent P and a classical P with local users

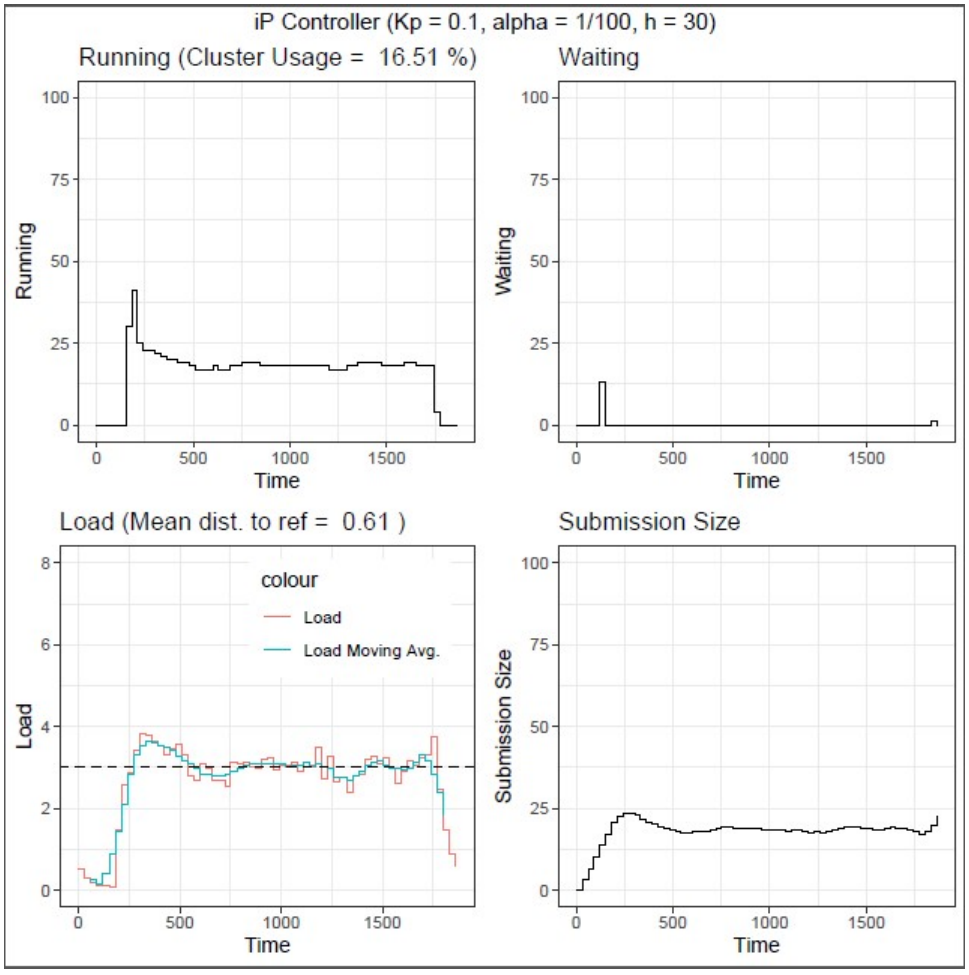


Fig. 11. Simulation results on the Grid'5000 using an Intelligent P based on cluster usage

4.2.2 Analysing the cluster utilisation of the MFC approach

Based on equation (8), we analysed the cluster utilisation performance metric of the MFC approach via an Intelligent P. A classical PI was also implemented in order to make a comparative analysis in this experimental set-up.

We considered the classical PI due to their good cluster utilisation property mentioned in [20], compared to the other approach carried out on this system. The results from our real time implementation on the Grid'5000 is shown in figure (12) and figure (11) respectively.

From the analysis in figure (12) and figure (11), the intelligent P improved the cluster usage by 16.51% while the classical PI showed a 15.81% improvement in the cluster usage.

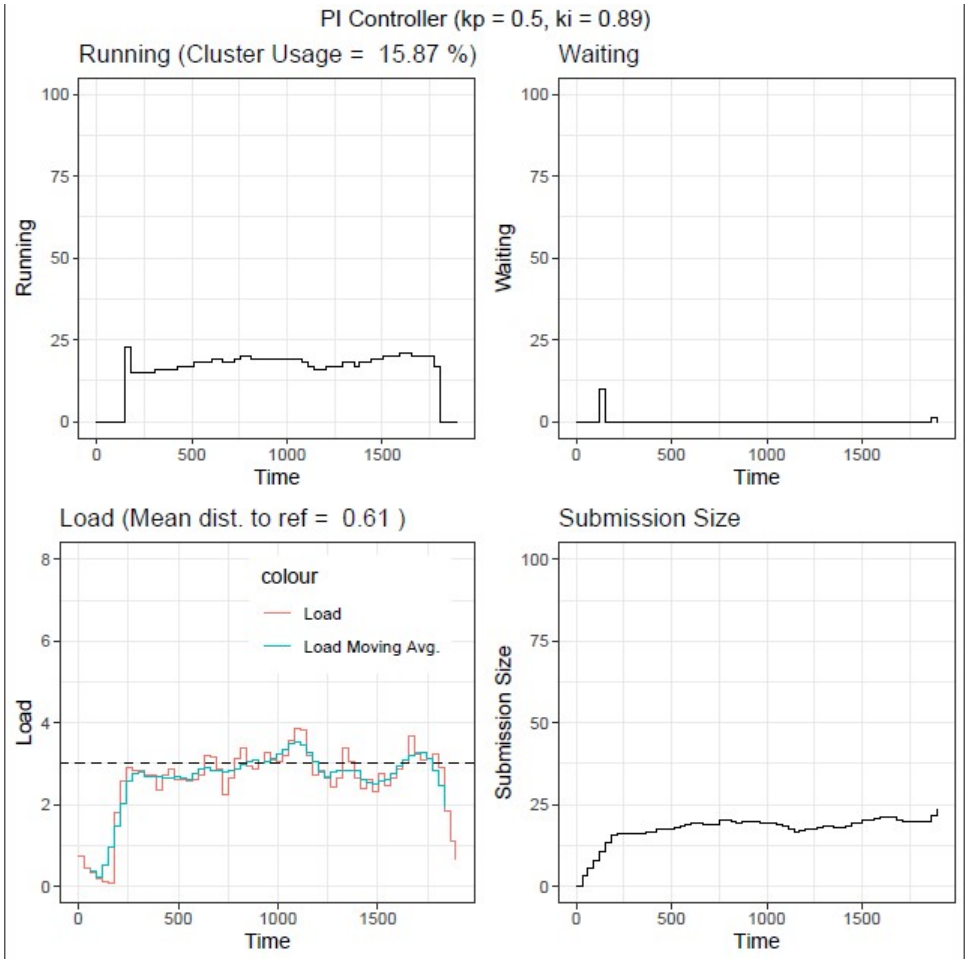


Fig. 12. Simulation results using a classical PI based on cluster usage

4.3 Results Discussion

In this section, we first run a validating simulation on the relation based on the gains between the classical and intelligent PID in order to establish a relational gain tuning technique. From the results in figure (4a) and (4b), we confirmed the equivalence between each of the control pairs based on their relational gains computation as proposed in [10].

Simulations on different plants were run using MATLAB Simulink based on the MFC approach, from the simulation results in figure (6), (5), (8), and (7), the MFC approach showed a very good reference tracking property, as well as robustness to disturbances which are key controller metrics in achieving our cluster resource management goals. These performance assurances as stated in the prospects of the Model free control (MFC) were confirmed from the simulation results.

From the experimental results on the Grid'5000 platform, in figure (10) and (9), we run experiments using the MFC approach via an intelligent Proportional control law, with and without local users. From the results, the MFC approach showed a better response time and robustness to external disturbances (when local users were considered) compared to the classical Proportional controller. The intelligent P presented a smooth tracking performance of our load set-point compared to the classical P over the experimental time duration.

We also observed an improvement in cluster usage in the real time simulation result using an intelligent P compared to a classical PI. The intelligent P showed 16.51% improvement in cluster usage compared to the 15.81% seen in the classical PI case.

Remark 7: We managed to achieve an increase in cluster resource usage without overloading the file-server using the intelligent P controller.

5 Conclusion and Perspective

Conclusion

In this work the model free control approach was adopted for the cluster computing resource management. This approach utilises an ultra-local model which is virtual and closes its loop via an intelligent Proportional-Integral-Derivative controller. The use of this approach indicates that, there is no need of a mathematical model to represent our system under study. Also this approach proposes a fast estimation technique to capture the unknown system dynamics and external disturbances subjected on the High performance computing(HPC) system application.

From the results, both in simulation and on the real time system, the Model Free Control approach guaranteed good reference tracking property and robustness to external disturbances in achieving our design objectives. The Model free control approach via the intelligent Proportional controller improved the cluster resource usage by 16.51 without overloading the file-server, compared to the 15.81% noticed in the classical Proportional-Integral controller.

Perspective

In relation to this work being the first data- driven control theory approach in the High performance computing (HPC) systems application, indeed a significant progress towards the objective of this research has been achieved. However, there is still a lot more to explore in relation to the data-driven control theory methods in the context of the cluster resource management.

In future works, a more adaptive tuning of the controller gains can be integrated in this approach to make it smarter and intuitive. An additional MFC controller can be implemented in the workflow to handle other system objectives such as energy consumption etc.

Also, with the encouraging results from this work, it validates the stimulating prospects machine learning techniques and that of control theory promises when merged in the management of cluster resource management. In this light, future research works can be tailored to explore this research domain.

APPENDIX

MFC IMPLEMENTATION IN MATLAB SIMULINK AND LIVE SCRIPT

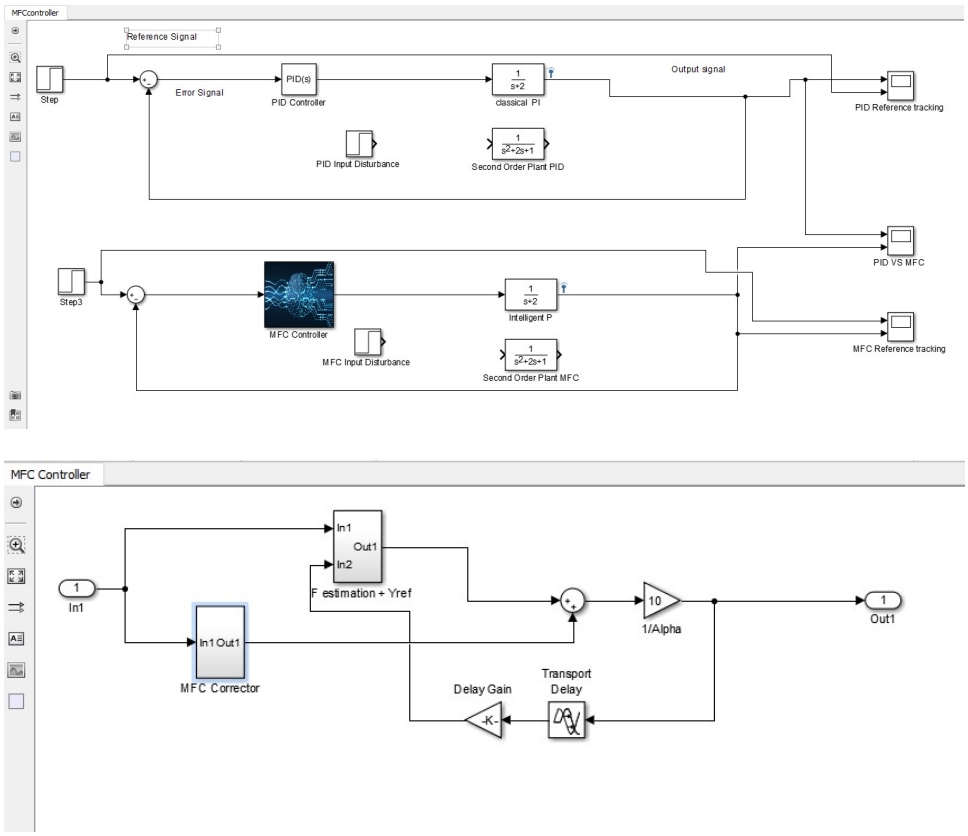


Fig. 13. The MFC Simulink blocks

```
clc;clear;close all
```

```
%Build time vector
tic                 % start timer to calculate CPU time
desired = 1;        % desired output, or reference point
dt=0.01;            % Sampling time
t=0:dt:20;

% Building the Plant Model
denum=[ 2 1];
num=[ 1 ];
%denum=[1 3 1];
%num=[ 1 ];
Plant_Model=tf(num,denum);
Discretized_Plant= c2d(Plant_Model,dt,'tustin')  % Discretized Plant Model using the Tustin method
```

This Plant Model can then be written in the form: $y_k = u_k + bu_{k-1)} + ... + b_{k-n}u_{k-n} + ay_{k-1} + ... + a_{k-n}y_{k-n}$. ...(1)

Discretized_Plant.variables='z^-1'

```
Discretized_Plant.variable=('z^-1')
```

Check the Plant model a response with a step

```
step(Discretized_Plant)
```

Check the Plant model a response with a step

```
step(Discretized_Plant)
```

NB: To write the Plant model in the form of equation (1) yields, $y_k = 0.002494u_k + 0.002494u_{k-1} + 0.995y_{k-1}$ .....(2)

Now here we build our control laws or Algorithms.

1. First we build the PID Corrector variants

2. Then we build the MFC Controller

* **Let's us first revise on the control laws Algorithms**

For the PID it takes the form: $u = K_p \cdot e_k + K_d \cdot \dot{e}_k + K_i \cdot \int e$ .....(3) **NB:** In the discrete domain, you will need adopt the Algorithm appropriately

For the MFC corrector takes the form: $u_k = u_{k-1} + \frac{(y_{ref} - y_{out} + PID)}{a}$ .....(4) **NB**: We will implemented it's discrete form in the next sections, I used this in building the controller in simulink.

```
%% Here we build the two control Algorithms to be used in equation (2)

% Predefine our variables.
Desired=1;              % Plant's Reference
b1=0.002492;            % First coefficient of the  u terms in the Plant model
b2=0.002492;            % Second coefficient of the u terms in the Plant Model
a=0.995;                % Second coefficient of the y term in the Plant Model
Alpha=100;             % The MFC tuning Parameter
dt=0.01;                % Sampling time
kp=30.90;                % Proportional gain
ki=2.10;                % Integral gain
kd=0;                   % Derivative gain
Time = 40;              % total simulation time in seconds
n = round(Time/dt);     % number of samples


%% Predfine the size of the input/output vectors

% Predefine the PID I/O Arrays
u_PID(1:n+1) = 0;                    % PID control law
PID_Output(1:n+1) = 0;               % Plant's Output using the PID control law
Error(1:n+1) = 0;                    % Error with respect to reference tracking
CumError(1:n+1)=0;                   % Cumulative PID error
Desired(1:n+1)=1;
```

```
%% We run the control Algorithm in a loop
 i;
    % Computing the Errors in both PID and MFC cases
     Error(i+1) = Desired(i) - Sys_PID_Output(i)   ;    % Error entering the PID controller
     MFC_Error(i+1) = Desired(i) - Sys_MFC_Output(i);     % error entering the MFC controller

     CumError=CumError + Error(i+1);
     MFC_CumError=MFC_CumError + MFC_Error(i+1);

     %% Building the Classical PID
     u_PID(i+1)=kp*Error(i+1) + ki*CumError*dt + kd.*(Error(i+1)-Error(i))/dt;  % PID control law in dicrete time

     %% Building the MFC PID
     MFC_PID(i+1)=kp*MFC_Error(i+1) + ki*MFC_CumError*dt + kd.*(MFC_Error(i+1)-MFC_Error(i))/dt;

     %% Building the MFC controller

     u_MFC(i+1)=kd/(Alpha*dt)*MFC_Error(i+1) + kp/(Alpha*dt)*MFC_CumError*dt + (-1)/(Alpha*dt)*(MFC_Error(i+1)-MFC_Error(i))/dt;    %% equation 28 from MFC Fliess
     u_MFC2(i+1)= 1/Alpha.*(Alpha.*u_MFC2(i) + (Desired(i+1)-Sys_MFC_Output(i+1)) + MFC_PID(i+1)); % MFC Control Law in discrete

     %% Building the Plant model with a PID
     Sys_PID_Output(i+1)= b1*u_PID(i+1) + b2*u_PID(i) + a*Sys_PID_Output(i);  % The Plant response when agumented with a PID

     %% Building the Plant model with an MFC controller
     Sys_MFC_Output(i+1)=b1*u_MFC2(i+1) + b2*u_MFC2(i) + a*Sys_MFC_Output(i);    % The Plant response when agumented with MFC


end
```

Fig. 14. The MFC Algorithm

References

[1] Sébastien Andary et al. "A dual model-free control of underactuated mechanical systems, application to the inertia wheel inverted pendulum". In: *2012 American Control Conference (ACC)*. IEEE. 2012, pp. 1029–1034.

[2] Maria Bekcheva et al. "Improving resource elasticity in cloud computing thanks to model-free control". In: *arXiv preprint arXiv:1810.03702* (2018).

[3] C Biscarat and B Bzeznik. "Synergy between the CIMENT tier-2 HPC centre and the HEP community at LPSC in Grenoble (France)". In: *Journal of Physics: Conference Series*. Vol. 513. 3. IOP Publishing. 2014, p. 032008.

[4] Romain Bourdais, Michel Fliess, and Wilfrid PERRUQUETTI. "Towards a model-free output tracking of switched nonlinear systems". In: *IFAC Proceedings Volumes* 40.12 (2007), pp. 504–509.

[5] Yuriy Brun et al. "Engineering self-adaptive systems through feedback loops". In: *Software engineering for self-adaptive systems*. Springer, 2009, pp. 48–70.

[6] bzizou. "The CiGri Middleware". In: Accessed on: 18/06/2021 09:21. 2013. URL: `http://ciment.univ-grenoble-alpes.fr/cigri/dokuwiki/doku.php?id=start`.

[7] Brigitte d'Andréa-Novel et al. "A mathematical explanation via "intelligent" PID controllers of the strange ubiquity of PIDs". In: *18th Mediterranean Conference on Control and Automation, MED'10*. IEEE. 2010, pp. 395–400.

[8] Rogério De Lemos et al. *Software Engineering for Self-Adaptive Systems III. Assurances: International Seminar, Dagstuhl Castle, Germany, December 15-19, 2013, Revised Selected and Invited Papers*. Vol. 9640. Springer, 2018.

[9] Michel Fliess and Cédric Join. "Delta hedging in financial engineering: Towards a model-free approach". In: *18th Mediterranean Conference on Control and Automation, MED'10*. IEEE. 2010, pp. 1429–1434.

[10] Michel Fliess and Cédric Join. "Model-free control". In: *International Journal of Control* 86.12 (2013), pp. 2228–2252.

[11] Michel Fliess, Cédric Join, and Hebertt Sira-Ramirez. "Complex continuous nonlinear systems: their black box identification and their control". In: *IFAC Proceedings Volumes* 39.1 (2006), pp. 416–421.

[12] Michel Fliess, Cédric Join, and Hebertt Sira-Ramirez. "Non-linear estimation is easy". In: *International Journal of Modelling, Identification and Control* 4.1 (2008), pp. 12–27.

[13] Michel Fliess et al. “Vers une commande multivariable sans mod\ele”. In: *arXiv preprint math/0603155* (2006).

[14] Yiannis Georgiou, Olivier Richard, and Nicolas Capit. “Evaluations of the lightweight grid cigri upon the grid5000 platform”. In: *Third IEEE International Conference on e-Science and Grid Computing (e-Science 2007)*. IEEE. 2007, pp. 279–286.

[15] Jeffrey O Kephart and David M Chess. “The vision of autonomic computing”. In: *Computer* 36.1 (2003), pp. 41–50.

[16] Ioan D Landau et al. “Can Adaptive Feedforward Control Improve Operation of Cloud Services?” In: *2018 26th Mediterranean Conference on Control and Automation (MED)*. IEEE. 2018, pp. 1–9.

[17] Mediawiki. “The Grid’5000; Grisou Cluster”. In: Accessed on: 19/06/2021 19:21. 2021. URL: `https://www.grid5000.fr/w/Nancy:Hardware`.

[18] Emmanuel Stahl et al. “Towards a control-theory approach for minimizing unused grid resources”. In: *Proceedings of the 1st International Workshop on Autonomous Infrastructure for Science*. 2018, pp. 1–8.

[19] Jorge Villagra and Carlos Balaguer. “A model-free approach for accurate joint motion control in humanoid locomotion”. In: *International Journal of Humanoid Robotics* 8.01 (2011), pp. 27–46.

[20] Agustın Gabriel Yabo et al. “A control-theory approach for cluster autonomic management: maximizing usage while avoiding overload”. In: *2019 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2019, pp. 189–195.