Technical report: A Systematic Survey on the Design of Self-Adaptive Software Systems using Control Engineering Approaches

Tharindu Patikirikorala, Alan Colman, Liuping Wang and Jun Han

***Abstract*—Control engineering approaches have been identified as a promising design tool to integrate self-adaptive capabilities into software systems. Introduction of the feedback loop and controller to the management system enables the software systems to achieve the runtime performance objectives and maintain the integrity of the system when they are operating in unpredictable and dynamic environments. There is a large body of literature that proposed control engineering solutions for different application domains, dealing with different performance variables and control objectives. In addition, the relevant literature is scattered over different conference proceedings, journals and research communities. Consequently, conducting a survey to analyze and classify the existing literature is a challenging task. In this paper we present the results of a systematic survey, which includes classification and analysis of 158 papers in the existing literature. In order to capture the characteristics of the control problems and solutions proposed in these papers we introduce a taxonomy. All the selected papers are classified according to this taxonomy and then quantitative survey results are presented. In addition, the trends and limitations, challenges and possible solutions of existing works are listed as well. Further, a set of design patterns harvested during this survey is covered as well, that may assist the design of control systems for self-adaptive systems in the future.**

***Index Terms*—Self-adaptive systems, control engineering, feedback control, performance management**

I. Introduction

In [141] Shaw compares the suitability of software engineering methodologies with the control engineering methodologies to design a cruise control system and further states that "... When the execution of a software system is affected by external disturbances forces or events that are not directly visible to or controllable by the software this is an indication that a control paradigm should be considered for the software architecture ...".Many state of the art software systems have become complex and large scale and have to deal with unpredictable environmental conditions and dynamics that cannot be sufficiently modeled by existing software engineering methodologies. Consequently, including [141] and many papers afterwards (e.g.,[137], [15], [45], [42]) have identified control engineering methodologies as a promising tool to implement self-adaptive software systems. Basically, control engineering methodologies, integrates the feedback loop and controller to the management system enabling to achieve the operational goals reducing the administration costs, while reacting to unpredictable disturbances and un-modeled system dynamics in a timely and effective manner. Further, rigorous mathematical foundation and well-established formal design tools in control engineering provide a systematic design process with the capabilities to analyze the validity and stability of the implemented management systems.

Motivated by these capabilities, many control solutions have been proposed for software systems. However, these efforts are scattered over different conference proceedings, journals and research communities and relates to different application domains, deals with different performance variables, control objectives and implements versatile control schemes. Analyzing and drawing common patterns form such a large body of literature is a challenging task, consequently results of the exiting surveys are significantly limited to certain application domains or solution domains. This paper overviews a details of a systematic survey and its results. The main objectives of this systematic survey are to (1) build a classification model of the existing literature, (2)

T. Patikirikoral, A. Colman and J. Han are with the Faculty of Information and Communication technology, Swinburne University of technologies, Australia.

L. Wang is with Faculty of Electrical and Computer Engineering,Royal Melbourne Institute of Technology, Australia

find out widely adapted modeling, control schemes and (3) harvest design patterns which could aid the development of self-adaptive systems in the future (4) list the trends, limitations and challenge of existing works and then propose possible solutions to investigate in the future. To achieve these objectives firstly, 158 papers are selected from different conference proceedings and journals. Secondly, a taxonomy is developed in order to capture the knowledge about each of these papers and then classify them in a systematic way. Finally, we present the results of the survey with a quantitative analysis and set of design patterns harvested during the survey. The classifications of the papers according to taxonomy provides the details based on the existing literature to select the suitable control system design variables and schemes for a particular application domain, to implement new control system for research or industrial software systems (hence, the code name '*horses for courses*').

The rest of the paper is organized as follows: Section II overviews the related work. The details of the survey methodology is presented in Section III. The taxonomy derived in this work and the survey results are covered in Section IV and V respectively. Finally, Section VI provides the concluding remarks.

II. Related Work

The surveys [137], [53], [21] analyze the attempts based on software engineering (generally focusing on architectural reconfiguration) to implement self-adaptive systems, giving less emphasis to control engineering methodologies. The detailed summery of several control engineering solutions proposed by several researchers can be found in [188], [48], [3]. In addition, lists of challenges and design rules when applying control engineering methodologies for software systems are presented in [23], [47], [45], [65], [64], [90].

There are several surveys related to this work that provide overviews of literature in different prospective and classifications. One of the initial surveys that is related to this work is [5], which covers applications of feedback control in web servers, network, scheduling and storage management. However, this survey does not include many works published after 2003 in this area. In [43], a comprehensive survey has been conducted on different types of control engineering approaches applied for middleware (e.g., web and application severs). A limited set of key research work that used different types of control system designs to manage performance in software systems is presented in [182]. In addition, [42] provides a classification of limited set of papers according to the performance attribute controlled by the control system. Brun et al in [15] also provides a calcification based on the non-adaptive and adaptive control system designs for software systems. Further, our work in [128] provides a detailed classification of literature according to the control system design technique used. Moreover, many of the papers (e.g., [55], [161], [168], [127]) included in this surveys provides detailed related work sections, however limited to their area of study.

The design patterns are well known in software engineering, in particular useful when systems are designed based on object-oriented programming (OOP). Several design patters to implement self-adaptive software systems based on OOP can be found in [39], [135].

In contrast to the above work, this systematic survey provides a comprehensive classification of the literature based on a taxonomy which includes the application area, dimensionality and controlled performance attributes of the target software system and the scheme, architecture, dimensionality and so on of the control system designed. In addition, we classify the literature based on the validation technique used. Furthermore, the patterns harvested during this survey are significantly different from the patterns in [39], [135], because these patterns are related to control engineering techniques, in contrast to OOP techniques.

III. Review Method

Kitchenham et al. in [72] provides a set of guidelines to conduct a systematic literature review, which includes the steps of formulating the research questions to be answered by the review and developing a review protocol. These guidelines are followed for instance in [27], [73], [152] to conduct systematic reviews in different research areas of software engineering. In this work, we also follow the guidelines in [72]. The details of the steps followed in this survey are listed in following subsections.

A. Research Questions

Formulation of the research questions is the main driving force of a systematic review [72]. The research questions addressed by this survey are:

RQ1: How can we classify the existing approaches based on characteristics of the target software system (problem domain) and control system implemented (solution domain)?

RQ2: What are the methods used to model the dynamics of the software system?

RQ3: What are the control schemes, control system architectures and controllers (algorithms) used by the existing work?

RQ4: What are the design patterns exist in these proposed control approaches?

B. Review Protocol

Developing a review protocol is important, in order to select, organize and analyze the existing work without the possibility of bias. The review protocol is a planed set of activities [72], which includes the following steps.

1) Search Process: The basic idea behind this step is to decide on search strings and sources to search for the relevant papers (so called primary studies) for a survey. Deciding search strings for this survey was challenging because there is a large literature spanning the areas of software and control engineering. In order to maintain the count of the search results manageable, based on our previous experience we selected the conferences and journals listed in Table I as sources. Further, *feedback control* and *QoS* were used as the search strings and well known literature search engines like IEEE explore, ACM Digital library, ScienceDirect and DBLP Computer Science Bibliography were used to assist and narrow down the search process.

The papers gathered from this process were further investigated to improve the coverage by including the papers cited in the selected paper and other papers that cited the selected paper. In addition, all the (total of 424) papers cited the text book [48] were included. Further, we included all papers that cited in other surveys related to feedback control or self-management systems (e.g., [42], [137], [15], [53], [47], [43]). Primarily the title, keywords and abstract of the paper were used to make a decision on the relevance of the paper. However, when it was inadequate to make a decision, the introduction and approach sections of the paper were reviewed as well.

2) Inclusion, Exclusion Criteria and Quality Assessment: The selected papers from the search process is further evaluated in this step to determine whether the selected studies are relevant in answering the research questions and meet the expectations of the study.

Inclusion criteria: As a basic inclusion criteria, the date of publication and language was used. All the papers published between 1^{st} of January 2000 to 1^{st} of November 2011 and written in English were included in this survey. Then, one of the major inclusion criteria was problem domain and solution domain. The papers that addressed the problems on automating the management of the software applications, middleware or environments that deployed software components (e.g. data centers) and the papers that propose solution to these problems based on control engineering methodologies were included. In order to be a control engineering solution, we investigated two major steps of control system design, i.e., modeling the dynamics of the system and controller implementation. The list of control engineering methodologies selected are covered in details in Section IV-B4. There were many papers that duplicated the same contributions in different papers or cases where the conference papers were extended to journal articles. Removing such duplications was a major challenge to avoid the bias of the systematic survey. In such cases, we included the most complete paper (e.g., journal version was included as suppose to the conference paper).

Exclusion criteria: There is a vast literature, which has used control engineering methodologies to automate management of mobile, wireless and routing networks. These studies are out of the scope of this survey. The papers that proposed management systems without utilizing control theoretic approaches (e.g., optimization solution) were also excluded. This also excludes the control solutions primarily based on fuzzy logy, neural networks, case based-reasoning and reinforcement learning. In addition, the papers that only deal with hardware (e.g, processor chips) or operating system level management issues with control solutions were also excluded. Further, there are many papers that present the challenges, design guidelines and short surveys (e.g.,

TABLE I: Conference proceedings and journals selected to gather papers

Source	Acronym
IEEE Transactions on Parallel and Distributed Systems	PDS
IEEE Transactions on Network and Service Management	NSM
International Conference on Autonomic Computing	ICAC
International Conference on Parallel and Distributed Systems	ICPADS
International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks	FeBID
International Workshop on Quality of Service	IWQoS
International Symposium on Software Engineering for Adaptive and Self-Managing Systems	SEAMS
International Conference on High Performance Computing	HiPC
International Conference on High Performance Computing and Communications	HPCC

[47], [188], [65], [95]) that were excluded because they do not meet the major inclusion criteria.

Quality assessment: Assessing the quality of the paper or its contributions is a challenging and complex task. In order to evaluate the quality of the selected papers we used following criteria

QA1: Is the paper peer-reviewed?

QA2: Does the paper provide a validation for the proposed solution?

If the answer to both these questions is '*yes*', we included the paper in this survey.

At the end of this step, 158 papers that met the above criteria were finalized as the primary studies of this survey.

3) Data Extraction: The next step is to finalize the data extraction strategies. In order to answer the research questions formulated in Section III-A, information has to be extracted from the selected papers accurately without any bias. The extracted data provides an abstract view and knowledge about a specific paper. To extract the data in a systematic and standardized way, we started off with a basic taxonomy, which was further developed during the data extraction process. The basic taxonomy was developed by the authors from their previous experience, which was sufficient to answer the aforementioned research questions. The details of the final taxonomy is presented in Section IV.

Firstly, a *data extraction form* was documented based on the taxonomy (see Figure 1). This document also included other details of the paper such as the title, authors, conference/journal, publication year, bibliography identifier and additional notes. Then, each paper was read in details and the data extraction form was filled by a one author, while another author rechecked the accuracy of the data extraction. When, there is a disagreement, both authors get-together in a discussion to reach a final decision. After the data extraction froms of all the papers were completed, a relational database schema was designed to record the data in a database management system. This was done to improve the accuracy and tractability of the classification and Meta-analysis tasks involved in the next steps of the survey by using the standard feature rich query languages provided by the database management system (For this purpose we used Microsoft SLQ server). Next, all the information in the data extraction forms was included in the relation database.

4) Synthesis of the Extracted Data: Final step is to analyze the recoded data and answer the research questions and present the results of the survey. The details of the outcomes of this step are presented in Sections IV and V.

IV. Taxonomy

This section presents the taxonomy we developed after the detailed analysis of the literature. This taxonomy provides a mechanism to extract the knowledge about a particular paper and represent it in a high-level of abstraction. It is also a tool to classify and mine patterns exists in different levels fo the hierarchy. The finalized taxonomy is shown in Figure 1. The first level of the taxonomy captures the characteristics of the *Target system*, *Control System* and *Validation* in the paper. These components also have different subcomponents. The final hierarchy of the taxonomy was developed by further refining the classifications during the data extraction in order to keep the taxonomy in a manageable size. This was done by removing or merging some subcomponents to others or adding new subcomponents which were not covered by the basic taxonomy we started off with. The details of these components are as follows:

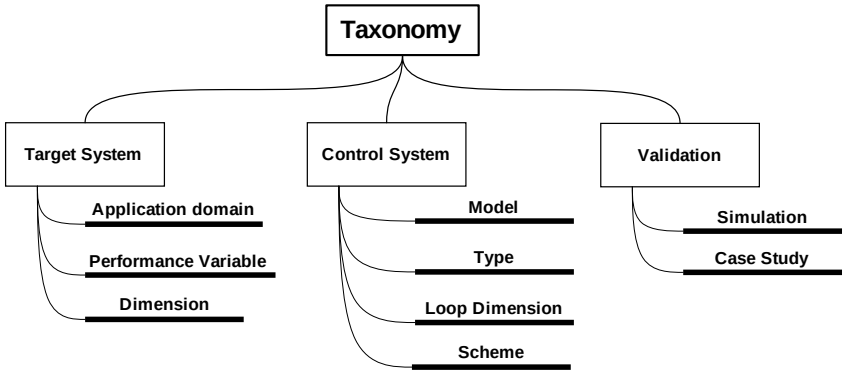


Fig. 1: The high-level sturcture of the taxonomy

A. Target system

This component represents the characteristics of the software system controlled by the proposed control engineering solution in each primary study. It was further classified by the subcomponents, which includes *Application domain*, *Performance/controlled variables* and *Dimension* of the target system. The application domains extracted from the selected papers include data center, virtual machine environments, data storage, middleware and real-time systems. The control engineering solutions are primary used to maintain the performance attributes at desired levels. Consequently, the performance/controlled variables of the target software system are major property that has to be investigated. The performance variables listed in Table II were used to classify the existing work. It is worth emphasizing is that all these variables are average measurements within a certain time period. The dimension of the target system relates to the control objectives of the problem at hand. It can be classified as the single-input-single-output (SISO) or multi-input-multi-output (MIMO), which represent a single control objective or multiple control objectives respectively.

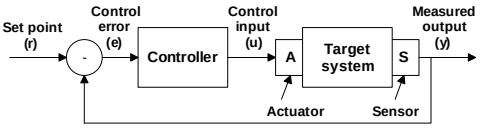
B. Control system

This component captures the knowledge about the control engineering solution proposed. The design and implementation of a control solution mainly includes two steps. Firstly, the behavior of the target system has to be modeled. Secondly, a suitable control system has to be implemented [48]. The details of the design and implementation process are captured in the following subcomponents.

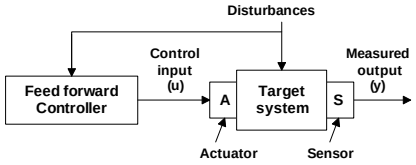
TABLE II: The list of performance variables

Performance variable	Definition
Response time	[102]
Throughput	[100]
Progress/Miss ratio	[103]
Power utilization	[161]
Processor utilization	[168], [103]
Hit rate/ratio	[107]
Memory	[51]
Queue length	[124]
Server utilization	[4], [26]
Tardiness	[187]
Number in system	[46]
Scheduling error	[6], [118]
Temperature	[32], [34]
Bandwidth	[52]
Failure rate	[94]
Performance degradation	[88]
Repetition Length	[30]
Benefit	[146]
Estimated weight	[14]

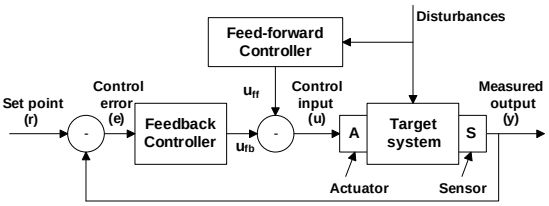
1) Model: The behavior of a target system can be formally represented by the analytical (first-principle) or black-box models. The analytical models represent the behavior of the system by using the underlying physical laws governing the target system (for instance, mass-balance, electrical, friction laws). However, in the case of software systems, such models are not available or significantly complex [48]. From this survey, we observed use of both of these techniques, however queuing models are also widely used to model the behavior of many different systems. Consequently, the queuing model was included as a classification uder this component. In contrast, the black-box models describe the system behavior with the input and output variables considering the system as a black-box. System identification (SID) is a widely used method



(a) Feedback control system



(b) Feed-forward control system



(c) Feedback and Feed-forward control system

Fig. 2: Block diagrams of different types of control systems

to construct the black-box model of a system. A SID experiment is conducted offline by applying a specially designed input signal on the system and to gather output data for a sufficient period of time. Then the gathered measurements of input and output data is used to estimate the model (typically, as a linear time invariant model) [48].

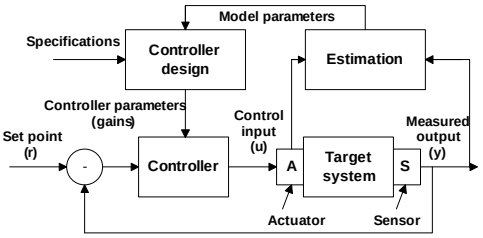
2) Type: **Feedback control system:** Figure 2a shows a block diagram of a feedback control system. The target system provides a set of performance variables referred to as *measured outputs* or simply *outputs*. Sensor monitors the outputs of the target system, while the *control inputs* or simply *inputs* can be adjusted through actuator to change the behavior of the system. The feedback controller is the decision making unit of the control system. The main objective of the controller is to maintain the outputs of the system sufficiently close to the desired values, by adjusting the inputs under disturbances. This desired values is translated in control system terms as the *set point signals*, which gives the option for the control system designer to specify the goals or values of the outputs that have to be maintained at runtime. The feedback control system is a reactive decision making mechanism, because it waits untill a disturbance affects the outputs of the system to make the necessary decisions.// **Feed-forward control system:** In contrast to feedback control, feed-forward control (See Figure 2b) measures the major disturbances and adjusts the inputs before the disturbance affects the system outputs. Consequently, it is considered as a proactive control mechanism. However, if the disturbance cannot be modeled accurately the performance of the feed-forward controller may be significantly poor. Further, typically in the cases where all the disturbances cannot be measured or modeled, the control objectives of maintaining the outputs around the set points (so-called set point tracking) may not be achieved.// **Feedback and Feed-forward control system:** Figure 2c shows the architecture of combined feedback and feed-forward control system. It addresses the limitations of both schemes, where the feed-forward control adjust the inputs based on disturbances that is measurable, while feedback control implements the set point tracking under unmeasured disturbances.

3) Loop Dimension: The design of a control system depends on the control objectives, i.e., whether it needs to achieve a single objective (SISO) or multiple objectives (MIMO). In the case of SISO system a SISO control system is sufficient to achieve the objectives. When there are multiple control objectives the control system that needs to be designed is complex. We observed mainly two solutions in our survey, including design of multiple-SISO control systems/loops or a MIMO controller. A multiple-SISO control system decomposes the multiple control objectives into multiple single objectives and then designs multiple SISO control systems. In contrast, the MIMO control system, achieves all the objectives using a single controller.

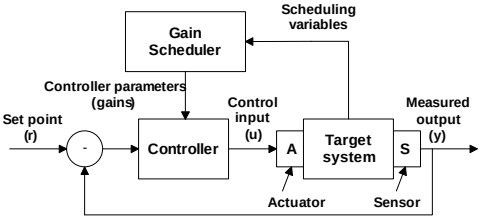
4) Scheme: This survey indicated that different control schemes have been used to implement the self-managing capabilities into software systems. We further, classify these schemes as basic and complex schemes. The difference is that the complex schemes are conceptual schemes typically realized using a single or multiple basic/complex control schemes.

*a) **Basic schemes**:*

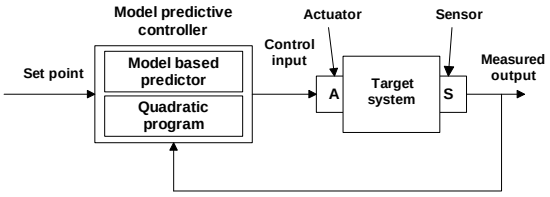
Fixed-gain control: The structure of a fixed-gain control scheme is same to that of Figure 2a. For instance, different variations of the Proportional Integral Derivative (PID) controller is used in exiting work as fixed gain controllers due to their robustness against modeling errors, disturbance rejection capabilities and simplicity [48]. The control algorithm



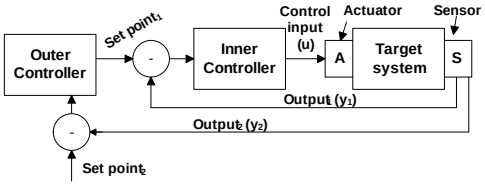
(a) Self-tuning adaptive control



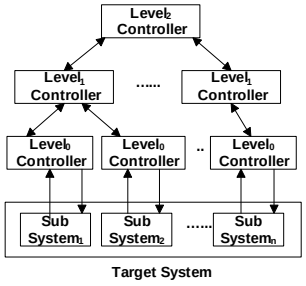
(b) Gain scheduling control



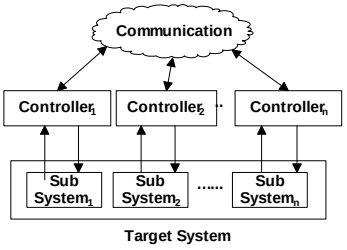
(c) Model predictive control



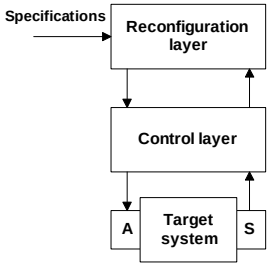
(d) Cascaded control



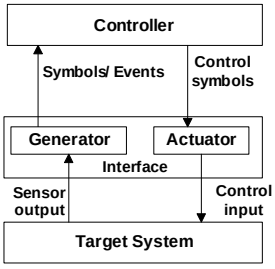
(e) Hierarchical control



(f) Decentralized control



(g) Reconfiguring control



(h) Hybrid control

Fig. 3: Block diagrams of different feedback control schemes

of PID controller is shown in equation (1)

$$u(k) = K_p e(k) + K_i \sum_{j=1}^{k} e(j) + K_d(e(k) - e(k-1)), \tag{1}$$

where $u(k)$ is the input for the current sample instance k, $e(k)$ is the different between output and set point and K_p, K_i and K_d are the parameters of the controller called *gains*. These gains are computed based on the model and other design specifications, however remain fixed at runtime (consequently, the name, **fixed gain** controller).

Adaptive control: In contrast to fixed gain control, adaptive control dynamically estimates the model parameters and gains of the controller at runtime. As shown in Figure 3a, adaptive controllers have a parameter adjustment loop, which derives these required parameters at runtime [147]. The parameters of the target system's model are estimated by the *Estimation* component, while the *Controller design* component uses these estimated model parameters and high-level control objectives provided by the designer to compute the gains of the controller.

Linear Quadratic Regulator (LQR): LQR is a optimal control strategy particularly useful in MIMO control system design. It uses a cost function, which represents a quadratic formula involving control error and control effort. The basic idea is to minimize the cost function so that the error is minimized with a small control effort. It also gives the opportunity to trade-off between speed of response to disturbances and overeating to noisy output signals [48]. For more details refer [48].

Model predictive control (MPC): MPC is a class of control algorithms that perform on-line optimization with a natural ability to deal with the system constraints and its design framework is entirely based on MIMO. It is similar to LQR, however the general idea behind MPC is to optimize the future behavior of the system outputs by computing the trajectory of the control inputs. Firstly, using the model of the system and the feedback (output) signals, the behavior of the system outputs is predicted over $k + N_p$, where k is the current time sample and N_p is called the *prediction horizon*. Then the objective of the predictive control is to maintain the predicted future outputs sufficiently close to the desired set point subject to various constraints on

input, output or combination of them that have to be optimized within the prediction horizon. The, MPC computes a sequence of inputs $u(k)$ to $u(k+N_c)$ to achieve the specified control objectives, where N_c is called the *control horizon*. However, only the first control input $u(k)$ will be implemented on the system in the current time sample, while discarding the rest of the sequence according to the *receding horizon control principle* [153]. The same process continues in the next sample intervals by sliding the prediction horizon one time step ahead while incorporating the feedback signals. For more details on MPC see [153]. The main components of the MPC system are shown in Figure 3c. MPC needs the system model and a standard quadratic programming solver to solve the optimization (or constraint) problem online.

*b) **Complex schemes**:*

Cascaded (nested) control: Most of the approaches assume that the set point specified in the controller remains constant or changes infrequently. The main objective of cascading control (Figure 3d) mechanism is to change the set point of the inner loop. The outer loop tries to maintain a one output around the set point by the mapping control objective into the inner loop control problem. Depending on the control error of the outer loop, it generates the set point periodically for the inner loop. When inner loop achieves its new set point, the control objectives of the outer loop will be indirectly achieved at the same time.

Gain scheduling: Gain scheduling is also regarded as an adaptive control mechanism in [147]. Figure 3b shows the block diagram of a gain scheduling control system. Here, predefined rules are implemented in the gain scheduling component depending on the prior knowledge about performance variables, disturbances and conditions. At runtime when the rules are satisfied the relevant controller gains are updated in the controller by the gain scheduling component. In contrast to adaptive control, gain scheduling does not have a model estimation component. Instead, it uses a predefined logic/rule based evaluation to change the controller online.

Reconfiguring control: In the adaptive control schemes the controller algorithm and the organization of the components in the loop stays fixed overtime [111], [130]. For different operating conditions and disturbances different control algorithms or loop organizations may provide better control [142], [130]. Reconfiguring control scheme is a conceptual approach with the main idea to change the control algorithms, models and architecture of the control system to deal with the changing operating regions of the target system. Figure 3g illustrates the conceptual layered architecture of reconfiguration control. The *control layer* consists of the control system (including the controller) providing the control in the current time instance. The responsibility of the *reconfiguration layer* is to reconfigure the architecture of the control layer (e.g., by changing controller) so that the control objectives of the target system can be achieved under requirement or environmental changes.

Hierarchical control: Figure 3e shows a general architecture of the hierarchical control scheme. The hierarchical control schemes can be used to realize control objectives of large distributed systems. The main idea is to implement divide-and-conquer concept, where low level (Level_0) controllers manage the sub systems of a large system, while high-level controllers act as a coordination layer of the lower level control systems. For instance, high-level controllers may adjust the control objectives of the lower level controllers after looking at system-wide control objectives.

Decentralized control: In contrast to the hieratical control system where the management decisions flow downwards from a centralized management entity, decentralized control manages each subsystem with a controller. There is no centralized entity that looks at the global control objectives and specifies the management objectives. The communication layer, on the other hand provides the information about the global state variables or just the states of the neighboring sub systems. Then, utilizing the local and information from the communication layer, each individual controller provides control in an independent manner. Consequently, the system-wide objectives are achieved in a decentralized fashion.

Hybrid control: Many software systems shows combined event and time based dynamics. All the above control schemes deals with discrete time based dynamics of software systems. The idea behind a hybrid control system is to incorporate both the event and time based dynamic aspects into control system design. The Figure 3h shows a basic architecture of a hybrid control system. The *interface* receives information about the variables as sampled

data from the target system, which then be converted to events/symbols by the generator when the special conditions are met. Depending on the symbols, the controller makes the control decisions to achieve the control objectives. The controller operates with a target system model, typically described by a finite automata (hybrid automata) which has a finite set of states and transition conditions between states. It starts from the starting state and move through different states (also referred as operating modes) depending on the events generated by the generator. Corresponding to the state, the system is treated as a discreet/continuous time system which is described by difference/differential equations that will be used to come up with the control decisions. The implementation of the controller or supervisory system is based on the finite automate theory, where a language is formed with the states and events. The set of states are grouped as *unsafe* states which the controller has to avoid. Then, given the current state, the controller can generate a trajectory of states avoiding the unsafe states to achieve the control objectives. The controller decisions are sent as control symbols to the actuator, which converts them to control inputs that can be applied in the target system.

As mentioned, the complex control schemes can be designed using basic schemes. In such cases we classify the paper in both subcomponents. Further, some papers introduced control solutions, which included multiple control shames together. In such cases as well we classified the paper under relevant subcomponents.

C. Validation

This component represents the type of validation provided in the paper to show the effectiveness of the proposed control engineering solution. It was further classified in to validation based on a *Simulation* or *Case study*. A simulation based validation relies on some kind of a simulation model of a target system and then implementing the proposed solution on it. To develop a simulation model well established techniques like discrete-event simulations or off-the-shelf simulation tools (e.g., Matlab) can be utilized. The case study based validations includes implementing a target system close to the real world settings and deploying the system in a physical environment. Then, that system is used to validate the control solution proposed. We further analyzed this subcomponent by extracting information on the case studies that used the benchmark software applications and workload generators. This was done to identify the characteristics of the benchmark and workload generators used and to provide further design guidelines and properties for such implementations in the future, which would aid the researchers to provide comprehensive validations.

V. Survey results

In this section we summarize the results of the survey. Firstly, we cover the general statistics of the papers based on the publication venue and year (Section V-A). Secondly, a quantitative analysis and classification of the existing literature is presented based on the taxonomy (Section V-B). Thirdly, the design patterns harvested during this survey will be listed (Section V-F). Lastly, in the discussion section (Section V-G), trends and challenges and the limitations of this survey will be summarized.

A. General statistics

In this section we provide statistics based on the publication venue and the year of the publication.

Table III provides an overview of the conferences and journals used as the publication venue of the papers included in this survey. In total, we recorded 77 distinct publication venues, indicating a significant fragmentation of the literature according to the publication venue. However, apart from the 12 venues listed in Table III, rest of the venues included less than 3 papers. Apart from the venues focusing on the software engineering or systems, the conferences which are primarily related to control engineering such as ACC and CDC have been used as publications venues as well. These statistics indicates the scattering of existing work in difference publication venues and research communities, which may have inhibited to conduct a systematic survey so far.

The statistics based on the publication year is illustrated in Table IV. It shows an increasing trend, indicating that the number of work which used control engineering solutions to solve the management problems in software systems has increased. A significant increment can be seen after year 2005. This may be because of the popularity of large scale cloud computing environments during that

TABLE III: Statistics based on the Publication venue

Publication venue	Number of papers	Percentage
ICAC	15	9.5
ACC	10	6.3
PDS	9	5.7
FeBID	8	5.1
RTAS	7	4.4
IWQOS	7	4.4
ICDCS	5	3.2
ECRTS	5	3.2
CDC	5	3.2
Computers	4	2.5
NSM	4	2.5
RTSS	4	2.5

TABLE IV: Statistics based on the publication year

Year	Number of papers
2000	1
2001	3
2002	10
2003	8
2004	8
2005	19
2006	17
2007	16
2008	19
2009	23
2010	20
2011	14

time period and afterwards, which led to many research challenges in automating management of such large scale systems. It can be further justified by the statistics of the work related to data center and VMs classified by the taxonomy. 100% of works that related to these two application domains have been done after year 2005. From these statistics we can speculate that the applications of control engineering methodologies have shown promise and may increase in the future as well.

B. Statistics Based on the Taxonomy

The main focus of this subsection is to answer the research questions **RQ2**, R**Q3** and **RQ4** formulated in sections III-A. The tables V, VII and X provide a quantitative analysis of components of the taxonomy focused in this paper. Further, tables VI, VIII, IX and XI groups the references of the papers according to subcomponents of the taxonomy.

TABLE V: Quantitative results of the subcomponents of 'Target system' component

Application Domain		
	Number of papers	Percentage
Middleware	54	31.4
Real-time systems	37	21.5
Data center	34	19.8
VM	24	14
Data Storage	22	12.8
Other	1	0.6
Performance variable		
	Number of papers	Percentage
Response time	75	37.9
Processor Utilization	40	20.2
Power Utilization	19	9.6
Progress/Miss ratio	17	8.6
Throughput	12	6.1
Hit rate/ratio	7	3.5
Queue length	5	2.5
Memory	4	2
Server utilization	4	2
Temperature	3	1.5
tardiness	2	1
Number in system	2	1
Scheduling error	2	1
Bandwidth	1	0.5
Failure rate	1	0.5
Performance degradation	1	0.5
Repetition Length	1	0.5
Benefit	1	0.5
Estimated weight	1	0.5
Dimension		
	Number of papers	Percentage
MIMO	95	60
SISO	63	39.9

C. Analysis of 'Target system' component

Application Domain: Table V indicates that the control theoretic applications are widely adapted to manage middleware (e.g., web servers, application servers and business process engines). Similarly, a large amount of (close to 20% of the papers) management problems involved with real-time systems and data center has been solved by control engineering solutions. A large amount of (close to 20% of the papers) management problems related to real-time systems and data center has been solved by control engineering solutions. In the case of real-time systems, another interesting observation was all the control solutions are proposed to manage soft-deadlines as suppose to hard-deadlines in unpredictable environments. The main reason for this observation is under unpredictable disturbances, the deadlines of some tasks could be violated, which is not tolerated in hard real-time systems. Over 10% of

TABLE VI: Classification of paper references according to the application domain and performance variable

	Data center	VM	Data Storage	Middleware	Real-time systems	Other
Response time	[190], [57], [142], [121], [165], [80], [120], [166], [81], [161], [154], [20], [50], [82], [163], [38], [79], [172], [189], [168]	[190], [85], [121], [165], [120], [166], [81], [161], [174], [51], [97], [164], [82], [163], [162]	[41], [101], [58], [17], [123], [24], [151], [28], [69]	[128], [179], [98], [145], [70], [186], [75], [185], [8], [67], [169], [102], [140], [106], [122], [132], [92], [130], [49], [19], [131], [66], [31], [175], [68], [96], [74], [76], [170], [56], [12], [129], [133], [150], [55], [114], [26]	[59], [58], [110], [83]	
Throughput	[86], [120]	[86], [120], [100], [40]	[115]	[70], [87], [66], [68], [176], [16], [114]		
Progress/Miss ratio		[125], [127], [126]			[183], [63], [149], [178], [10], [104], [144], [62], [60], [184], [171], [9], [93], [103]	
Power Utilization	[80], [81], [161], [167], [20], [156], [134], [82], [79], [84], [157]	[81], [100], [161], [117], [82], [162], [40]	[18]	[40], [56]	[143]	
Processor Utilization	[190], [121], [165], [89], [166], [167], [134], [177], [7], [189], [168], [91]	[190], [121], [165], [166], [51], [177]	[119], [89]	[54], [8], [19], [36], [148], [22]	[183], [160], [63], [104], [144], [105], [139], [78], [29], [60], [159], [184], [171], [93], [32], [33], [180], [158], [103], [112]	
Hit rate/ratio			[109], [108], [77], [107], [185], [37], [173]			
Memory		[51]		[25], [36], [22]		
Queue length	[155]	[155]		[2], [44], [13], [124]		
Server utilization				[136], [1], [4], [26]		
Tardiness					[187], [61]	
Number in system				[46], [71]		
Scheduling error					[6], [118]	
Temperature	[34]				[32], [33]	
Bandwidth			[52]			
Failure rate					[94]	
Performance degradation					[88]	
Repetition Length						[30]
Benefit			[146]			
Estimated weight					[14]	

the papers have investigated the management issues of data storage and virtual machines (e.g., databases, memory and cache) environments as well.

Performance variables: Table V lists the performance variables of the target systems controlled by the control solutions proposed in the primary studies. 21 different performance variables were identified. The first 12 attributes have been used in more than 1 paper, while the rest have not been used widely. From the statistics the response time is one of the major performance attributes investigated in the existing literature. The reasons for this could be that the response time is (1) the user perceived performance attribute of the system (2) one of the attribute specified in SLAs and (3) useful to formulate a set point tracking control problem. Although, throughput is also considered as a main performance variable, it is difficult to be used when a set point tracking problem is needed to be formulated. This is because, throughput generally varies with workload rate linearly till it saturates, consequently, specifying a constant set point is difficult. The processor utilization is one of the other performance variables looked at by a large number of papers. The increasing cost and demand of power has become a major issue in data center operations, thus controlling or reducing power utilization has gained attention in the past few years [79], [157]. It is also encouraging to see that the power management is also looked at in 10% of the papers. In contrast, many of the other performance attributes are utilized in less than 10% of the papers. It is also evident that many variables related to queuing models are also used as the performance variable (e.g., queue length, server utilization and number in system). However, the issue with such attributes is coming up with desirable values as set points, which would indirectly achieve the main performance variables (e.g., response time) interested by the users of software systems.

Diemention: The simple classification based on the target system dimensions indicates that most of the target software systems are MIMO systems (60% of the papers, compared to 30% classified under SISO systems). It indicates that there are typically multiple control objectives in the management problem of a software system (See Table V).

Table VI, groups the papers, based on the application domain and performance variable controlled. Some papers belong to more than one cell of the table, because they deal with multiple control objectives or MIMO systems. This classification provides interesting characteristics of which performance variables to monitor and manage depending on the application domain (horses for cources). For instance, response time is one of the main performance variables utilized in domains such as data centers, VM environments and middleware. In contrast, the processor utilization and miss ratio are the performance variables managed in the real-time system domain. Similarly, power and processor utilization have been used to compose the management objectives of the data centers and VM environments. It is also evident that the selection of the performance variable highly depends on the application domain. Some of performance variables have no relevance in particular domains (e.g., hit rate in VM environments).

TABLE VII: Quantitative results of the subcomponents of 'Control system' component

Model		
	Number of papers	Percentage
Black box	105	64.8
Queuing	30	18.5
Analytical model	27	16.7
Type		
	Number of papers	Percentage
Feedback	139	88
Feedback + forward	17	10.8
Feed-forward	2	1.3
Loop Dimension		
	Number of papers	Percentage
SISO	71	44.4
MIMO	50	31.3
Multi-SISO	39	24.4
Scheme		
	Number of papers	Percentage
Fixed	63	28.8
Adaptive	33	15.1
LQR	25	11.4
MPC	24	11
Hierarchical	17	7.8
Gain scheduling	16	7.3
Cascade	14	6.4
Hybrid	11	5
Reconfiguring	10	4.6
Decentralized control	6	2.7

D. Analysis of 'Control system' component

Model: As discussed in Section IV-B1, the three major modeling techniques used in existing literature were classified in to black-box, queuing theoretic and analytical models. From the statistics

TABLE VIII: Classification of paper references according to the modeling mechanism and type of control system

	Queuing	Black-box	Analytical model
Feedback	[155], [57], [71], [80], [44], [81], [136], [154], [13], [132], [92], [1], [82], [79], [56], [12], [55]	[190], [183], [128], [179], [121], [98], [146], [107], [25], [70], [119], [41], [86], [30], [101], [87], [89], [120], [52], [59], [58], [81], [17], [185], [100], [161], [18], [187], [61], [109], [108], [63], [8], [67], [174], [178], [167], [102], [104], [144], [123], [122], [117], [132], [143], [92], [125], [130], [139], [24], [19], [99], [62], [131], [151], [115], [60], [20], [66], [51], [36], [156], [134], [68], [6], [97], [50], [164], [4], [28], [163], [127], [177], [38], [37], [181], [171], [126], [84], [173], [9], [129], [69], [124], [7], [22], [189], [168], [162], [40], [150], [114], [91], [93], [32], [33]	[155], [94], [46], [2], [88], [80], [110], [160], [10], [105], [78], [29], [148], [159], [82], [184], [118], [79], [34], [157], [133], [32], [158], [83], [103], [112]
Feed-forward	[172]	[142]	
Feedback + forward	[85], [186], [75], [165], [166], [140], [106], [175], [96], [74], [170], [26]	[40], [165], [166], [49], [31], [96], [76], [180]	[49]

TABLE IX: Classification of paper references according to the control system dimension and type of control scheme

	SISO	Multi-SISO	MIMO
Fixed	[94], [46], [40], [98], [54], [145], [88], [119], [71], [75], [101], [87], [89], [52], [44], [59], [58], [17], [110], [187], [136], [174], [10], [122], [117], [143], [125], [151], [31], [4], [28], [37], [176], [74], [76], [84], [172], [12], [124], [91]	[121], [186], [185], [109], [8], [167], [102], [104], [140], [106], [123], [92], [60], [51], [175], [159], [170], [77], [7], [14]	[161], [159], [162], [32]
Adaptive	[179], [85], [98], [107], [25], [108], [6], [127], [96], [126], [173], [69], [168]	[121], [165], [166], [171], [9]	[70], [41], [86], [120], [100], [67], [99], [115], [66], [68], [97], [38], [181], [40], [180]
MPC	[131], [1], [56]	[154], [13], [159]	[155], [57], [80], [81], [161], [18], [160], [105], [29], [156], [159], [82], [163], [184], [79], [157], [129], [32], [158]
LQR	[164]		[41], [86], [120], [100], [61], [178], [24], [19], [99], [62], [20], [66], [36], [68], [97], [50], [164], [38], [181], [34], [22], [40], [114], [180], [83]
Reconfiguring	[128], [142], [149]	[183], [104], [60], [171], [103]	[146], [162]
Cascade		[165], [166], [63], [8], [144], [139], [51], [134], [77], [16], [189], [93], [33], [112]	
Gain scheduling	[30], [169], [132], [130], [49], [131], [177], [133], [168], [150], [55]	[167], [134], [189]	[146], [18]
Hierarchical		[190], [121], [165], [166], [63], [167], [144], [134], [93]	[57], [86], [80], [120], [81], [161], [82], [157]
Decentralized control		[154], [144], [93]	[155], [160], [180]
Hybrid	[2], [78], [148], [1], [56]		[57], [80], [81], [82], [118], [79]

the black-box models are more popular than first principle models. Close to 65% of the papers have utilized black-box models, because of the difficulty of constructing a first principle model to represent the system dynamics and runtime behavior. In contract, the queuing models are used to incorporate proactive control to the solutions, utilizing the feed-forward mechanism. With these statistics we can conclude that the black-box models have been more useful to capture the dynamics of software systems and then design successful control systems.

Type: Apart from the two papers which used stand-alone feed-forward control loop, close to 99% the papers have included a feedback control loop. The main reason for this statistic is that accurate measurements of the disturbances faced by the target software systems are hard to acquire. Many papers that used feed-forward control loop used workload rates as the primary disturbance. This is in fact true in most cases, however it is hard to measure the workloads because of the stochastic nature of the workloads faced by the software systems. In addition, there are other un-modeled disturbances such as garbage collections, compiler optimizations and competition for resources between components that would affect a performance of the feed-forward control loop. As a consequence, the feedback loop has been used in most cases to achieve the desired control objectives (set point tracking) under un-modeled dynamics. Table VIII, indicates feedback control has been implemented based on black-box models. In addition, the feed-forward control has been realized using the predictive qualities of queuing models. In particular, in the case of the feedback combined feed-forward control systems, feed-forward component is designed using queuing model, while the feedback loop is designed based on a black-box model. Therefore, the model classification under feedback- feed-forward control type illustrates similar clustering under queuing and black-box models.

Loop dimension: The dimension of the controller or the control loop also reveals interesting results (see Table VII). Many control solutions proposed in the literature so far deals with a single control objective. 71 papers in total designed SISO control solutions. However, 89 papers have looked at MIMO control problems and proposed Multi-SISO or MIMO control solutions for them. Our statistics further shows that close to 80% of the MIMO control solutions were proposed in the last 5 years.

TABLE X: Quantitative results of the subcomponents of 'Validation' component

Validation Method	Number of papers	Percentage
Simulation	53	33.5
Case study/test bed	114	72.2

Scheme: From Table VII, it is evident that many papers (63 in total) have utilized fixed gain (PID control variations) in the control solutions. The reason for this may be the simplicity and robustness of that control scheme. Most of the PID controllers (64%) were used to design SISO control systems. In contrast, MIMO control systems were designed with control schemes like MPC and LQR. The reason is that MPC and LQR are naturally designed to deal with MIMO control problems. This is further illustrated in Table IX, where most of the MIMO control systems are clustered under MPC and LQR schemes. The adaptive control, MPC and LQR schemes are utilized more than 10% of the papers, indicating the usefulness of such control schemes to tackle control problems in software systems. Although, the scale, typical operating conditions and disturbances faced by software systems demand complex control solutions such as gain scheduling, hierarchical, cascade and reconfiguring control schemes, that have not been widely adopted compared to the basic control schemes.

E. Analysis of 'Validation' component

The proposed control approach in each paper has been validated basically either by simulation or case study based on a test bed (9 papers have utilized both) (see Table X). The case study based validation method looks like the widely adopted validation technique. However, the groupings of the papers in Table XI shows that control solutions proposed for real-time systems are validated using the simulation environments compared to other application domains. On the contrary, in all the other application domains case studies have been widely accepted as a validation technique.

In addition to the above statistics and classifications, we further analyzed the case studies that utilized or included benchmark software systems in their validation. It is noteworthy that only few papers either used or specified such usage of benchmarks in their paper. Table XII summarizes the

TABLE XI: Classification of paper references according to the application domain and validation provided

Validation	Simulation	Case Study
Data center	[57], [80], [167], [99], [20], [134], [163], [181], [34], [157]	[7], [38], [50], [57], [79], [81], [82], [84], [91], [89], [86], [154], [120], [121], [142], [155], [177], [172], [157], [156], [161], [190], [189], [163], [168], [166], [165]
VM	[163]	[40], [51], [81], [82], [85], [86], [117], [120], [121], [125], [126], [174], [127], [155], [177], [161], [190], [97], [162], [163], [164], [100], [166], [165]
Data	[119], [18], [28], [77]	[17], [41], [52], [69], [58], [89], [101], [115], [123], [146], [151], [108], [109], [24]
Middleware	[46], [2], [179], [71], [13], [132], [49], [131], [148], [1], [74], [76], [170], [133], [26]	[4], [8], [37], [16], [175], [25], [150], [36], [92], [44], [49], [179], [169], [40], [55], [66], [67], [68], [70], [56], [74], [75], [12], [98], [87], [102], [114], [19], [122], [124], [128], [129], [31], [136], [185], [140], [145], [130], [170], [176], [173], [96], [54], [107], [106], [22], [186]
Real-time systems	[94], [110], [160], [63], [178], [10], [143], [105], [78], [62], [29], [60], [159], [184], [171], [118], [9], [93], [33], [180], [158], [83], [103], [112]	[6], [14], [104], [183], [61], [58], [59], [88], [139], [144], [149], [159], [158], [32], [187]

TABLE XII: Quantitative results of the case studies that used a Benchmark

Benchmark	Modeled application	Number of papers
TPC-W [113]	Retail store	11
Rubis [138]	Auction site	9
Trade6[1]	Stock trading application	5
RUBBoS[2]	News forum	4

TABLE XIII: Quantitative results of the Workload generators used in case studies

Workload generator	Number of papers
httpref [116]	19
SURGE [11]	15
Benchmark workload generator	13
SEPC[3]	5
Apache Ab[4]	3
Apache Jmerter[5]	1

results of those papers. It lists the benchmarks that have been used more than one paper. These benchmark applications model and represent different software applications. Form the papers that used case studies 27 papers have used a single or multiple benchmarks in their validations. The TPC-W benchmark is used by 11 papers, where as less than 10 papers have used other case studies listed in Table XII. These benchmarks demonstrate different performance characteristics in different environments and workload patterns. Further, the workload patterns simulated by these benchmarks are also utilized in the validation, which stress different tiers in a multi-tier software system (e.g., browsing mix and transactional workload mix in Rubis benchmark). It is hard to reason or provide guidelines to which benchmark to use in a case study. The benchmarks in Table XII have been useful in validation of the control solutions proposed in the existing literature, so that can be used as a reference list to select a suitable benchmark for a case study in the future.

One of the important tools to provide case study based validation is the workload generator. Again, only some papers mentioned about the particular workload generator used in the validation. The statistics are summarized in Table XIII. The workload generators that do not based on any benchmark such as *httpref* and *SURGE* have been used in 34 papers. These workload generators can be used to evaluate the performance of the web servers using web workloads. These workload generators are categorized as open-loop workload generators which send requests without considering the completion of the previous requests send by a particular user. They also provide different tunable parameters to adjust user think times based on the selected probability distributions (e.g., SURGE). In contrast, the workload generator provided by the *Rubis* benchmark simulates close-loop workloads, i.e., The next request is sent based on the completion of the previous request of a particular user.

F. Design Patterns

A design pattern is a reusable solution to a common problem related to design of systems [35],

[135]. This section lists several design patterns harvested during this survey related to control system design for software systems. It is noteworthy that these patterns are significantly different to general OOP design patterns in software engineering. These patterns may be useful in future research and design of feedback control systems for industrial software systems.

The design patterns listed in this section are composed based on the common recurrent problems in papers included in this survey. During the detailed review of each paper, we identified a list of common problems and then documented the selected solutions. These were noted down in the special notes section of the data extraction form. Then using this information, the patterns were finalized in the step of synthesizing the results. In order to validate this analysis we also provide the percentages of the papers selected the specified solution to resolve the common problem.

Typically, a design pattern is documented using a template. Such templates to represent OOP design patters could be found in [35], [135]. However, all the elements in these templates are not directly useful to document the design patterns for a control solution. Instead, we use a template which includes the *Pattern name*, *Problem* (a short description about the problem), *Solution* (a short description about the solution), *Context* (where/when to apply) and *Known use and statistics* (papers that used this pattern). Using this template we now introduce the set of design patterns composed during this survey.

DP1

Pattern name - ARXmodelorder

Problem - In the case of black-box modeling of a software system, typically, autoregressive exogenous input (ARX) models are used [48]. The standard form of the ARX model is as follows:

$$y(k) = \sum_{i=0}^{n} a_i y(k-i) + \sum_{j=0}^{m} b_j u(k-d-j) \quad (2)$$

where, n and m are the order of the model, a_i and b_j are the parameters of the model, d is the delay (time intervals taken to observe a change of input in the output) and k stands for the current sample instance. The problem here is what is the order (n and m) of the ARX model to represent the dynamics of the software systems with sufficient accuracy.

Solution- First (n, m =1) or second order (n, m =2) models can be used to represent the dynamics sufficiently accurately to reduce the computational and design complexity.

Context - When ARX models are used to represent the system as a black-box.

Known use and statistics- We were able to extract this information from 70 papers. 68% of the papers have used first order ARX model, where as the rest have used second order models. We were not able to find any papers which used a third order model or higher.

DP2

Pattern name - TypeOfControlSystem

Problem - Out of the feedback, feed-forward and feedback + feed-forward control system types, which types to use? This problem was seen in papers that used feedback + feed-forward type control systems.

Solution- Depending on the dynamics of the software systems and environmental conditions, include a feedback loop as a part of the control solution.

Context- In any control system design for a software system, where all the disturbances that may affect the performance cannot be accurately measured. Further, the problem at hand should be a set point tracking problem.

Known use and statistics- Table VII lists the statistics for this pattern. Close to 99% papers have used feedback loop as an essential part of the solution.

DP3

Pattern name - PIDSelection

Problem- When a PID control is decided to be used, which components should be included in the controller (i.e., is it propositional (P), integral (I), derivative (D) terms or combination of former). The (P) term improves the settling time by reacting to the disturbances. The (I) term contributes to reach the set point and eliminate the steady state error. The (D) term on the other hand reduces the effect of overshooting, however is sensitive to noisy output signals.

Solution- Include (I) term and to improve the settling time after disturbances, (P) term can be included as well. Set derivative term to zero.

Context- If a PID controller is needed to be designed. Further, the problem at hand should be

TABLE XIV: Statistics of papers used PID control variations

PID variation	Number of papers	Percentage
PI	47	53.4
I	23	26.1
P	9	10.2
PID	9	10.2

a set point tracking problem.

Known use and statistics- Table XIV summarizes the statistics. It is evident that more than 89% of the papers that used a PID control scheme have included (I) term in their solution. At the same time 74% of the papers have used (P) term. In contrast, total of 9 papers have used (D) term, however together with (P) and (I) terms. It is evident that the PI controller has been widely used compared to any other variations of PID controller.

DP4

Pattern name - MIMOControllerSelection

Problem-When a MIMO control system is selected to be designed, which basic control schemes to use? Many papers opted to implement PID controller and ran into issues of tuning the controller due to large gains.

Solution- Selection of optimal controller, designed based on a cost function. From the control schemes listed in Section IV-B4, LQR and MPC belongs to optimal control category. Their designs naturally deal with MIMO systems.

Context- When a MIMO control system is designed for a MIMO target system.

Known use and statistics- From the 50 papers that proposed MIMO control solutions, 43 of them have used LQR (total of 24 papers) or MPC (total of 19 papers) in their designs.

DP5

Pattern name - HierarchicalCascade

Problem- In hierarchical control system design, how to convey the management decisions of the higher-levels to the lower levels.

Solution- The higher-level controller specifies the new control objectives at runtime to the lower level controller by adjusting the set points of the lower-level controllers using the cascade control design technique. (It is noteworthy that cascade control is a basic form of hierarchical control, however only deals with a single system.)

Context- When a Hierarchical control system design is required.

Known use and statistics- From the 17 papers that proposed different hierarchical control solutions, 35% have used cascade control systems as a basic building block.

DP6

Pattern name - OuterLoopTimePeriod

Problem- How to set the sample time periods of the inner and outer loop of a cascade control system. The main issue arise here is the coordination between two loops. If both loops operate in the same time intervals, the outer loop may not see the effects of the control decisions made by the inner loop. This is vital to the stability of the cascade control system, because outer loop may keep on changing the set point of the inner loop realizing that the inner loop has not achieved the objectives.

Solution- Set sample time period of the inner loop ¡ outer loop. The time difference between the sample periods has to be decided based on the analysis of the settling time of the inner loop to reach new set point.

Context- If the solution includes cascade control.

Known use and statistics- Selection of this solution was observed in all the papers that proposed cascade control (i.e., 100% of the papers).

DP7

Pattern name-HigherLevelTimePeriod

Problem- How to set the sample time periods of the higher and lower level controllers of a hierarchical control system. The reason for this issue is same as reason discussed in OuterLoopTimePeriod pattern.

Solution- Set the sample time period of level n ¡ level n+1 , where n = 0,1, ...

Context- If the solution includes a hierarchical control system design.

Known use and statistics- This pattern was observed in over 76% of the papers that used hierarchical control systems (e.g., [157], [159], [161], [190], [80], [82], [93], [57], [166], [165]).

DP8

Pattern name - DiscreteInput

Problem- Some of the inputs exposed by the target systems belong to a limited set of discrete values. For instance, the processor frequencies exposed

for dynamic voltage scaling belongs to a limited set. In such cases design of a linear control is difficult.

Solution- Use hybrid control strategies. As described in Section IV-B4, a state-space based search mechanisms are implemented to select the input from the input set which move the system to safe states or the set point.

Context- When the target system exposes inputs having limited set of discrete values.

*Known use and statistics-*The 9 out of 11 papers that proposed hybrid control, utilized such control schemes to solve the issue of discrete inputs, in particular papers [2], [78], [1], [80], [81], [82], [79], [56], [57] .

DP9

Pattern name -HypervisorCPUShedluer

Problem- Many state of the art virtualization platforms (e.g., Xen, VMwaer) provide *work- conserving* and *non-work-conserving* CPU scheduling modes [120]. In work-conserving, if a VM is not totally using the specified maximum CPU share, it can be allocated to another VM which demands more CPU. In non-work-conserving mode each VM can use only the maximum specified limit. What mode to use in the case of control system design?

Solution- The *non-work-conserving mode* (or cap based mode). This is because behavior of the work-conserving mode is hard to control and it is designed to achieve set of scheduling objectives at the CUP level without concerning about the performance objectives at the software system. For instance, if a control system comes up with CPU shares for two VMs after looking at the application level objectives of each VM, then these shares must be implemented precisely. Such guarantee is hard to achieve with work-conserving mode, which will induce so-called *input noise* in the control system, which would lead to instabilities (See [120] for experiment results). Further, the non-work-conserving scheduler provides better performance isolation between the VMs sharing the same CPU.

Context- When a virtualization platform is used to manage the CPU share of multiple VMs sharing dual or multi core CPU capacity. However, this pattern is equally true for other resources such as memory and network bandwidth.

Known use and statistics- This pattern was observed in over 80% (19 out of 24) of the papers that proposed solutions for CPU sharing between VMs to achieve performance objectives. For instance in [120], [121], [161], [100], [165], [126].

G. Discussion

In this section we list the trends (Section V-G1), limitations, challenges and future research directions (Section **??**) and the limitations of this survey (Section **??**),.

1) Trends: This section we analyze the trends in literature based on the taxonomy.

Application domain: There is an increasing trend to apply control engineering solutions for application domains such as data centers and virtual machine environments as suppose to middleware and storage systems. In fact, work on those areas have started after year 2004 and increased afterwards. This trend is maybe because of the popularity of utility computing model in the past few years.

Performance variables: The performance variables such as response time, process utilization and power utilization illustrate increasing trends as well. . Similarly, the complex SLAs with penalties composed based on the quality of service attributes such as response time may have affected theses trends as well. The costs of power and demands for green computing may have triggered the investigating control solutions to manage performance variables like power. Control system dimension: There are increasing trend of designing MIMO control solution compared to SISO and Multi-SISO control systems. This trend is encouraging to see because of the MIMO control systems tackle the multiple control objectives in an effective manner.

Control scheme: There are positive trends in application of fixed, adaptive, LQR and MPC control schemes. It is hard to decide on the trends of other complex control sachems because of lack of application (data). Although, we don't have precise statistics, we also observed that increasing use of Kalman filtering in control solutions (e.g., [79], [57], [172]). The Kalman filter is widely adopted to track state variables which required by the control solution in control engineering literature. Similar concept is used by some of the papers used in this survey to track workload rates using queuing models.

Validation: There is a clear increasing trend of using case studies for validation compared to simulations.

VI. Conclusions

Many self-adaptive software systems have been implemented based on the control engineering methodologies in the existing literature. This paper provides the details of a systematic survey of such control engineering approaches proposed in 158 papers in the literature. A classification model was built to capture and represent the information about each paper in a high-level abstraction. In addition, the quantitative results and set of design patterns harvested from this survey are also presented, which may provide helpful guidance when implementing control solutions for software systems in the future.

References

[1] S. Abdelwahed, N. Kandasamy, and S. Neema. Online control for self-management in computing systems. In *IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 368 – 375, may 2004.

[2] S. Abdelwahed, S. Neema, J. Loyall, and R. Shapiro. A hybrid control design for qos management. In *24th IEEE Real-Time Systems Symposium*, pages 366 – 369, dec. 2003.

[3] T. Abdelzaher, Y. Diao, J. L. Hellerstein, C. Lu, and X. Zhu. Introduction to control theory and its application to computing systems. In *International Conference on Measurement and Modeling of Computer Systems*, 2008.

[4] T. Abdelzaher, K. Shin, and N. Bhatti. Performance guarantees for web server end-systems: a control-theoretical approach. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):80 –96, jan 2002.

[5] T. Abdelzaher, J. Stankovic, C. Lu, R. Zhang, and Y. Lu. Feedback performance control in software services. *IEEE Control Systems*, 23(3):74 – 90, june 2003.

[6] L. Abeni, L. Palopoli, and G. Buttazzo. On adaptive control techniques in real-time resource allocation. In *Euromicro Conference on Real-Time Systems*, pages 129 –136, 2000.

[7] W. Aly and H. Lutfiyya. Using feedback control to manage qos for clusters of servers providing service differentiation. In *IEEE Global Telecommunications Conference*, volume 2, page 5 pp., nov.-2 dec. 2005.

[8] W. H. F. Aly and H. Lutfiyya. Dynamic adaptation of policies in data center management. In *Proceedings of the Eighth IEEE International Workshop on Policies for Distributed Systems and Networks*, pages 266–272, Washington, DC, USA, 2007. IEEE Computer Society.

[9] M. Amirijoo, P. Brannstrom, J. Hansson, S. Gunnarsson, and S. H. Son. *Toward Adaptive Control of QoS-Importance Decoupled Real-Time Systems*. 2007.

[10] M. Amirijoo, J. Hansson, S. Gunnarsson, and S. Son. Enhancing feedback control scheduling performance by on-line quantification and suppression of measurement disturbance. In *IEEE Real Time and Embedded Technology and Applications Symposium*, pages 2 – 11, march 2005.

[11] P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. In *Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, SIGMETRICS '98/PERFORMANCE '98, pages 151–160, New York, NY, USA, 1998. ACM.

[12] L. Bertini, J. Leite, and D. Mosse. Siso pidf controller in an energyef?cient multi-tier web server cluster for e-commerce. In *Workshop on Feedback Control Impl. and Design in Computing Systems and Networks*, FeBID '07, 2007.

[13] V. Bhat, M. Parashar, H. Liu, M. Khandekar, N. Kandasamy, and S. Abdelwahed. Enabling self-managing applications using model-based online control strategies. In *IEEE International Conference on Autonomic Computing*, pages 15 – 24, june 2006.

[14] A. Block, B. Brandenburg, J. H. Anderson, and S. Quint. An adaptive framework for multiprocessor real-time system. In *Proceedings of the 2008 Euromicro Conference on Real-Time Systems*, ECRTS '08, pages 23–33, Washington, DC, USA, 2008. IEEE Computer Society.

[15] Y. Brun, G. Marzo Serugendo, C. Gacek, H. Giese, H. Kienle, M. Litoiu, H. Müller, M. Pezzè, and M. Shaw. Software engineering for self-adaptive systems. chapter Engineering Self-Adaptive Systems through Feedback Loops, pages 48–70. Springer-Verlag, Berlin, Heidelberg, 2009.

[16] B. Chen, X. Peng, and W. Zhao. Towards runtime optimization of software quality based on feedback control theory. In *Proceedings of the First Asia-Pacific Symposium on Internetware*, Internetware '09, pages 10:1–10:8, New York, NY, USA, 2009. ACM.

[17] M. Chen, X. Wang, R. Gunasekaran, H. Qi, and M. Shankar. Control-based real-time metadata matching for information dissemination. In *Proceedings of the 2008 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 133–142, Washington, DC, USA, 2008. IEEE Computer Society.

[18] M. Chen, X. Wang, and X. Li. Coordinating processor and main memory for efficientserver power control. In *Proceedings of the international conference on Supercomputing*, ICS '11, pages 130–140, New York, NY, USA, 2011. ACM.

[19] M. Chen, X. Wang, and B. Taylor. Integrated control of matching delay and cpu utilization in information dissemination systems. In *International Workshop on Quality of Service*, pages 1 –9, july 2009.

[20] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing server energy and operational costs in hosting centers. *SIGMETRICS Perform. Eval. Rev.*, 33:303–314, June 2005.

[21] B. H. Cheng, R. Lemos, H. Giese, P. Inverardi, J. Magee, J. Andersson, B. Becker, N. Bencomo, Y. Brun, B. Cukic, G. Marzo Serugendo, S. Dustdar, A. Finkelstein, C. Gacek, K. Geihs, V. Grassi, G. Karsai, H. M. Kienle, J. Kramer, M. Litoiu, S. Malek, R. Mirandola, H. A. Müller, S. Park, M. Shaw, M. Tichy, M. Tivoli, D. Weyns, and J. Whittle. Software engineering for self-adaptive systems. chapter Software Engineering for Self-Adaptive Systems: A Research Roadmap, pages 1–26. Springer-Verlag, Berlin, Heidelberg, 2009.

[22] Y. Diao, N. Gandhi, J. Hellerstein, S. Parekh, and D. Tilbury. Using mimo feedback control to enforce policies for interrelated metrics with application to the apache web server. In *IEEE/IFIP Network Operations and Management Symposium*, pages 219 – 234, 2002.

[23] Y. Diao, J. L. Hellerstein, and S. Parekh. Control of large scale computing systems. *SIGBED Rev.*, 3:17–22, April 2006.

[24] Y. Diao, J. L. Hellerstein, A. J. Storm, M. Surendra, S. Lightstone, S. Parekh, and C. Garcia-Arellano. Incorporating cost of control into the design of a load balancing controller. *IEEE Real-Time and Embedded Technology and Applications Symposium*, 0:376, 2004.

[25] Y. Diao, X. Hu, A. Tantawi, and H. Wu. An adaptive feedback controller for sip server memory overload protection. In

Proceedings of the 6th international conference on Autonomic computing, ICAC '09, pages 23–32, New York, NY, USA, 2009. ACM.

[26] B. Du and C. Ruan. Modeling and robust control for trusted web server. In *Proceedings of the 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, EUC '10, pages 659–665, Washington, DC, USA, 2010. IEEE Computer Society.

[27] T. Dybå and T. Dingsøyr. Empirical studies of agile software development: A systematic review. *Inf. Softw. Technol.*, 50:833–859, August 2008.

[28] M. Eiblmaier, R. Mao, and X. Wang. Power management for main memory with access latency control. In *International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks*, 2009.

[29] Z. Fangling and W. Jinbiao. Lp based mpc algorithm in distributed real-time systems with end-to-end tasks. In *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-2 (CIMCA-IAWTIC'06) - Volume 02*, pages 1049–1054, Washington, DC, USA, 2005. IEEE Computer Society.

[30] N. Fescioglu-Unver and M. Kokar. Application of self controlling software approach to reactive tabu search. In *IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, pages 297 –305, oct. 2008.

[31] R. Fontaine, P. Laurencot, and A. Aussem. Mixed neural and feedback controller for apache web server. *ICGST International Journal on Computer Network and Internet Research, CNIR*, 09:25–30, December 2009.

[32] X. Fu, X. Wang, and E. Puster. Dynamic thermal and timeliness guarantees for distributed real-time embedded systems. In *IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 403 –412, aug. 2009.

[33] Y. Fu, N. Kottenstette, Y. Chen, C. Lu, X. Koutsoukos, and H. Wang. Feedback thermal control for real-time systems. In *IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 111 –120, april 2010.

[34] Y. Fu, C. Lu, and H. Wang. Robust control-theoretic thermal balancing for server clusters. In *IEEE International Symposium on Parallel Distributed Processing*, pages 1 –11, april 2010.

[35] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[36] N. Gandhi, D. Tilbury, Y. Diao, J. Hellerstein, and S. Parekh. Mimo control of an apache web server: modeling and controller design. In *American Control Conference*, volume 6, pages 4922 – 4927 vol.6, 2002.

[37] A. Gao, D. Mu, H. Su, and W. Pan. Proportional hit rate in caching service: A feedback control approach. In *International Symposium on Computer Network and Multimedia Technology*, pages 1 –4, jan. 2009.

[38] A. Gao, H. Zhou, Y. Hu, D. Mu, and W. Hu. Proportional delay differentiation service and load balancing in web cluster systems. In *INFOCOM IEEE Conference on Computer Communications Workshops*, pages 1 –2, march 2010.

[39] H. Gomaa and M. Hussein. Software reconfiguration patterns for dynamic evolution of software architectures. In *Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture*, pages 79–88, Washington, DC, USA, 2004. IEEE Computer Society.

[40] J. Gong and C.-Z. Xu. A gray-box feedback control approach for system-level peak power management. In *International Conference on Parallel Processing (ICPP)*, pages 555 –564, sept. 2010.

[41] A. Gounaris, C. Yfoulis, and N. Paton. An efficient load balancing lqr controller in parallel database queries under random perturbations. In *IEEE Control Applications, (CCA) Intelligent Control*, pages 794 –799, july 2009.

[42] J. Guitart, J. Torres, and E. Ayguadé. A survey on performance management for internet applications. *Concurr. Comput. : Pract. Exper.*, 22:68–106, January 2010.

[43] R. Gullapalli, C. Muthusamy, and V. Babu. Control systems application in java based enterprise and cloud environments a survey. *International Journal of Advanced Computer Science and Applications*, 2:103 –113, 2011.

[44] T. Heinis and C. Pautasso. Automatic configuration of an autonomic controller: An experimental study with zero-configuration policies. In *International Conference on Autonomic Computing*, pages 67 –76, june 2008.

[45] J. Hellerstein. Challenges in control engineering of computing systems. In *American Control Conference*, volume 3, pages 1970 –1979 vol.3, 30 2004-july 2 2004.

[46] J. Hellerstein, Y. Diao, and S. Parekh. A first-principles approach to constructing transfer functions for admission control in computing systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 3, pages 2906 – 2912 vol.3, dec. 2002.

[47] J. Hellerstein, S. Singhal, and Q. Wang. Research challenges in control engineering of computing systems. *IEEE Transactions on Network and Service Management*, 6(4):206 –211, december 2009.

[48] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury. *Feedback Control of Computing Systems*. John Wiley and Sons, 2004.

[49] D. Henriksson, Y. Lu, and T. Abdelzaher. Improved prediction for web server delay control. In *Euromicro Conference on Real-Time Systems, 2004*, pages 61 – 68, june-2 july 2004.

[50] J. Heo, P. Jayachandran, I. Shin, D. Wang, T. Abdelzaher, and X. Liu. Optituner: On performance composition and server farm energy minimization application. *IEEE Transactions on Parallel and Distributed Systems*, 22(11):1871 –1878, nov. 2011.

[51] J. Heo, X. Zhu, P. Padala, and Z. Wang. Memory overbooking and dynamic control of xen virtual machines in consolidated environments. In *IFIP/IEEE International Symposium on Integrated Network Management*, pages 630 –637, june 2009.

[52] H. Huang and A. Grimshaw. Automated performance control in a virtual distributed storage system. In *IEEE/ACM International Conference on Grid Computing*, pages 242 –249, 29 2008-oct. 1 2008.

[53] M. C. Huebscher and J. A. McCann. A survey of autonomic computingdegrees, models, and applications. *ACM Comput. Surv.*, 40:7:1–7:28, August 2008.

[54] Y. Jiang, D. Meng, J. Zhan, and D. Liu. Adaptive mechanisms for managing the high performance web-based applications. In *International Conference on High-Performance Computing in Asia-Pacific Region*, pages 6 pp. –397, july 2005.

[55] A. Kamra, V. Misra, and E. Nahum. Yaksha: a self-tuning controller for managing the performance of 3-tiered web sites. In *IEEE International Workshop on Quality of Service*, pages 47 – 56, june 2004.

[56] N. Kandasamy, S. Abdelwahed, and J. Hayes. Self-optimization in computer systems via on-line control: application to power management. In *International Conference on Autonomic Computing*, pages 54 – 61, may 2004.

[57] N. Kandasamy, S. Abdelwahed, and M. Khandekar. A hierarchical optimization framework for autonomic performance management of distributed computing systems. In *IEEE International Conference on Distributed Computing Systems*, page 9, 2006.

[58] K.-D. Kang, J. Oh, and S. Son. Chronos: Feedback control of a real database system performance. In *IEEE International Real-Time Systems Symposium*, pages 267 –276, dec. 2007.

[59] K.-D. Kang, J. Oh, and Y. Zhou. Backlog estimation and management for real-time data services. In *Euromicro Conference on Real-Time Systems*, pages 289 –298, july 2008.

[60] K.-D. Kang, S. Son, and J. Stankovic. Managing deadline miss ratio and sensor data freshness in real-time databases. *IEEE Transactions on Knowledge and Data Engineering*, 16(10):1200 – 1216, oct. 2004.

[61] W. Kang, S. Son, and J. Stankovic. Design, implementation, and evaluation of a qos-aware real-time embedded database. *IEEE Transactions on Computers*, PP(99):1, 2010.

[62] W. Kang, S. Son, J. Stankovic, and M. Amirijoo. I/o-aware deadline miss ratio management in real-time embedded databases. In *IEEE International Real-Time Systems Symposium*, pages 277 –287, dec. 2007.

[63] W. Kang, S. H. Son, and J. A. Stankovic. Dracon: Qos management for large-scale distributed real-time databases. *Journal of Software*, 4(7):747–757, 2009.

[64] C. Karamanolis, M. Karlsson, and X. Zhu. Designing controllable computer systems. In *Proceedings of the 10th conference on Hot Topics in Operating Systems*, pages 9–9, Berkeley, CA, USA, 2005. USENIX Association.

[65] M. Karlsson. Design rules for producing controllable computer services. In *Network Operations and Management Symposium*, pages 1 –14, april 2006.

[66] M. Karlsson. Maximizing the utility of a computer service using adaptive optimal control. In *International Conference on Networking, Sensing and Control*, pages 89 –94, 0-0 2006.

[67] M. Karlsson and M. Covell. Dynamic black-box performance model estimation for self-tuning regulators. In *International Conference on Automatic Computing*, pages 172–182, Washington, DC, USA, 2005. IEEE Computer Society.

[68] M. Karlsson and C. Karamanolis. Non-intrusive performance management for computer services. In M. van Steen and M. Henning, editors, *Middleware*, volume 4290 of *Lecture Notes in Computer Science*, pages 22–41. Springer Berlin / Heidelberg, 2006. 10.1007/119250712.

[69] M. Karlsson, C. Karamanolis, and X. Zhu. Triage: Performance differentiation for storage systems using adaptive control. *Trans. Storage*, 1:457–480, November 2005.

[70] M. Karlsson, X. Zhu, and C. Karamanolis. An adaptive optimal controller for non-intrusive performance differentiation in computing services. In *International Conference on Control and Automation, 2005*, volume 2, pages 709 –714 Vol. 2, june 2005.

[71] M. Kihl, A. Robertsson, and B. Wittenmark. Analysis of admission control mechanisms using non-linear control theory. In *IEEE International Symposium on Computers and Communication*, pages 1306 – 1311 vol.2, june-3 july 2003.

[72] B. Kitchenham and S. Charters. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

[73] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman. Systematic literature reviews in software engineering - a systematic literature review. *Inf. Softw. Technol.*, 51:7–15, January 2009.

[74] M. Kjaer, M. Kihl, and A. Robertsson. Resource allocation and disturbance rejection in web servers using slas and virtualized servers. *IEEE Transactions on Network and Service Management*, 6(4):226 –239, december 2009.

[75] M. Kjaer and A. Robertsson. Analysis of buffer delay in web-server control. In *American Control Conference (ACC)*, pages 1047 –1052, 30 2010-july 2 2010.

[76] M. Kjser, M. Kihl, and A. Robertsson. Response-time control of a single server queue. In *IEEE Conference on Decision and Control*, pages 3812 –3817, dec. 2007.

[77] B.-J. Ko, K.-W. Lee, K. Amiri, and S. Calo. Scalable service differentiation in a shared storage cache. In *International Conference on Distributed Computing Systems*, pages 184 – 193, may 2003.

[78] X. Koutsoukos, R. Tekumalla, B. Natarajan, and C. Lu. Hybrid supervisory utilization control of real-time systems. In *IEEE Real Time and Embedded Technology and Applications Symposium*, pages 12 – 21, march 2005.

[79] D. Kusic and N. Kandasamy. Risk-aware limited lookahead control for dynamic resource provisioning in enterprise computing systems. In *IEEE International Conference on Autonomic Computing*, pages 74 – 83, june 2006.

[80] D. Kusic, N. Kandasamy, and G. Jiang. Approximation modeling for the online performance management of distributed computing systems. In *International Conference on Autonomic Computing*, page 23, june 2007.

[81] D. Kusic, N. Kandasamy, and G. Jiang. Combined power and performance management of virtualized computing environments serving session-based workloads. *IEEE Transactions on Network and Service Management*, 8(3):245 –258, september 2011.

[82] D. Kusic, J. Kephart, J. Hanson, N. Kandasamy, and G. Jiang. Power and performance management of virtualized computing environments via lookahead control. In *International Conference on Autonomic Computing*, pages 3 –12, june 2008.

[83] D. Lawrence, J. Guan, S. Mehta, and L. Welch. Adaptive scheduling via feedback control for dynamic real-time systems. In *Performance, Computing, and Communications, 2001. IEEE International Conference on.*, pages 373 –378, apr 2001.

[84] C. Lefurgy, X. Wang, and M. Ware. Server-level power control. In *International Conference on Autonomic Computing*, page 4, june 2007.

[85] N. Leontiou, D. Dechouniotis, and S. Denazis. Adaptive admission control of distributed cloud services. In *2010 International Conference on Network and Service Management*, pages 318 –321, oct. 2010.

[86] Q. Li, Q.-f. Hao, L.-m. Xiao, and Z.-j. Li. An integrated approach to automatic management of virtualized resources in cloud environments. *Comput. J.*, 54:905–919, June 2011.

[87] Z. Li, D. Levy, S. Chen, and J. Zic. Auto-tune design and evaluation on staged event-driven architecture. In *Proceedings of the 1st workshop on MOdel Driven Development for Middleware*, MODDM '06, pages 1–6, New York, NY, USA, 2006. ACM.

[88] K. Liang, X. Zhou, K. Zhang, and R. Sheng. An adaptive performance management method for failure detection. In *Proceedings of the 2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pages 51–56, Washington, DC, USA, 2008. IEEE Computer Society.

[89] H. C. Lim, S. Babu, and J. S. Chase. Automated control for elastic storage. In *Proceeding of the 7th international conference on Autonomic computing*, ICAC '10, pages 1–10, New York, NY, USA, 2010. ACM.

[90] H. C. Lim, S. Babu, J. S. Chase, and S. S. Parekh. Automated

control in cloud computing: challenges and opportunities. In *Proceedings of the 1st workshop on Automated control for datacenters and clouds*, ACDC '09, pages 13–18, New York, NY, USA, 2009. ACM.

[91] H. C. Lim, S. Babu, J. S. Chase, and S. S. Parekh. Automated control in cloud computing: challenges and opportunities. In *Proceedings of the 1st workshop on Automated control for datacenters and clouds*, ACDC '09, pages 13–18, New York, NY, USA, 2009. ACM.

[92] H. Lin, K. Sun, S. Zhao, and Y. Han. Feedback-control-based performance regulation for multi-tenant applications. In *International Conference on Parallel and Distributed Systems*, pages 134 –141, dec. 2009.

[93] S. Lin and G. Manimaran. Double-loop feedback-based scheduling approach for distributed real-time systems. In T. Pinkston and V. Prasanna, editors, *High Performance Computing*, volume 2913 of *Lecture Notes in Computer Science*, pages 268–278. Springer Berlin / Heidelberg, 2003. 10.1007/978-3-540-24596-429.

[94] S. Lin and G. Manimaran. A feedback-based adaptive algorithm for combined scheduling with fault-tolerance in real-time systems. In L. Boug and V. Prasanna, editors, *High Performance Computing*, volume 3296 of *Lecture Notes in Computer Science*, pages 101–110. Springer Berlin / Heidelberg, 2005. 10.1007/978-3-540-30474-616.

[95] M. Litoiu, M. Woodside, and T. Zheng. Hierarchical model-based autonomic control of software systems. *SIGSOFT Softw. Eng. Notes*, 30:1–7, May 2005.

[96] X. Liu, J. Heo, L. Sha, and X. Zhu. Queueing-model-based adaptive control of multi-tiered web applications. *IEEE Transactions on Network and Service Management*, 5(3):157 –167, september 2008.

[97] X. Liu, X. Zhu, P. Padala, Z. Wang, and S. Singhal. Optimal multivariate control for differentiated services on a shared hosting platform. In *IEEE Conference o Decision and Control*, pages 3792 –3799, dec. 2007.

[98] X. Liu, X. Zhu, S. Singhal, and M. Arlitt. Adaptive entitlement control of resource containers on shared servers. In *IFIP/IEEE International Symposium on Integrated Network Management*, pages 163 – 176, may 2005.

[99] X. Liu, X. Zhu, J. Yao, Z. Wang, and S. Singhal. Reduced dimension control based on online recursive principal component analysis. In *International Workshop on Feedback Control Implementation and Design for Computing Systems and Networks*, 2008.

[100] Z. Liu and D. Mu. Coordinating power and performance in virtualized environments. In *IEEE International Conference on Computer Science and Automation Engineering*, volume 3, pages 705 –709, june 2011.

[101] C. Lu, G. A. Alvarez, and J. Wilkes. Aqueduct: Online data migration with performance guarantees. In *Proceedings of the 1st USENIX Conference on File and Storage Technologies*, FAST '02, Berkeley, CA, USA, 2002. USENIX Association.

[102] C. Lu, Y. Lu, T. Abdelzaher, J. Stankovic, and S. Son. Feedback control architecture and design methodology for service delay guarantees in web servers. *IEEE Transactions on Parallel and Distributed Systems*, 17(9):1014 –1027, sept. 2006.

[103] C. Lu, J. A. Stankovic, S. H. Son, and G. Tao. Feedback control real-time scheduling: Framework, modeling, and algorithms*. *Real-Time Syst.*, 23:85–126, July 2002.

[104] C. Lu, X. Wang, and C. Gill. Feedback control real-time scheduling in orb middleware. In *IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 37 – 48, may 2003.

[105] C. Lu, X. Wang, and X. Koutsoukos. Feedback utilization control in distributed real-time systems with end-to-end tasks. *IEEE Transactions on Parallel and Distributed Systems*, 16(6):550 – 561, june 2005.

[106] Y. Lu, T. Abdelzaher, C. Lu, L. Sha, and X. Liu. Feedback control with queueing-theoretic prediction for relative delay guarantees in web servers. In *IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 208 – 217, may 2003.

[107] Y. Lu, T. Abdelzaher, C. Lu, and G. Tao. An adaptive control framework for qos guarantees and its application to differentiated caching. In *IEEE International Workshop on Quality of Service, 2002*, pages 23 – 32, 2002.

[108] Y. Lu, T. Abdelzaher, and G. Tao. Direct adaptive control of a web cache system. In *American Control Conference*, volume 2, pages 1625 – 1630, 4-6, 2003.

[109] Y. Lu, A. Saxena, and T. Abdelzaher. Differentiated caching services; a control-theoretical approach. In *Distributed Computing Systems, 2001. 21st International Conference on.*, pages 615 –622, apr 2001.

[110] Z. Lu, J. Hein, M. Humphrey, M. Stan, J. Lach, and K. Skadron. Control-theoretic dynamic frequency and voltage scaling for multimedia workloads. In *Proceedings of the 2002 international conference on Compilers, architecture, and synthesis for embedded systems*, CASES '02, pages 156–163, New York, NY, USA, 2002. ACM.

[111] K. M. M, K. Baclawski, and Y. A. Eracar. Control theory-based foundations of self-controlling software. *IEEE Intelligent Systems*, 14(3):37–45, 1999.

[112] M. Maggio and A. Leva. Toward a deeper use of feedback control in the design of critical computing system components. In *IEEE Conference on Decision and Control*, pages 5985 – 5990, 2010.

[113] M. Marden. An architectural evaluation of java tpc-w. In *Proceedings of the 7th International Symposium on High-Performance Computer Architecture*, HPCA '01, pages 229–, Washington, DC, USA, 2001. IEEE Computer Society.

[114] V. Mathur, P. Patil, V. Apte, and K. Moudgalya. Adaptive admission control for web applications with variable capacity. In *International Workshop on Quality of Service, 2009*, pages 1 –5, july 2009.

[115] A. Merchant, M. Uysal, P. Padala, X. Zhu, S. Singhal, and K. Shin. Maestro: quality-of-service in large disk arrays. In *Proceedings of the 8th ACM international conference on Autonomic computing*, ICAC '11, pages 245–254, New York, NY, USA, 2011. ACM.

[116] D. Mosberger and T. Jin. httperfa tool for measuring web server performance. *SIGMETRICS Perform. Eval. Rev.*, 26:31–37, December 1998.

[117] R. Nathuji, P. England, P. Sharma, and A. Singh. *Feedback driven qos-aware power budgeting for virtualized servers*. 2009.

[118] Y. Niu and G. Dai. Reservation-based state feedback scheduler for hybrid real-time systems. In *IEEE International Conference on High Performance Computing and Communications*, pages 198 –204, sept. 2008.

[119] J. Oh and K.-D. Kang. An approach for real-time database modeling and performance management. In *Proceedings of the 13th IEEE Real Time and Embedded Technology and Applications Symposium*, pages 326–336, Washington, DC, USA, 2007. IEEE Computer Society.

[120] P. Padala, K.-Y. Hou, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, and A. Merchant. Automated control of multiple virtualized resources. In *Proceedings of the 4th ACM Euro-*

pean conference on Computer systems, EuroSys '09, pages 13–26, New York, NY, USA, 2009. ACM.

[121] P. Padala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, and K. Salem. Adaptive control of virtualized resources in utility computing environments. In *Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, EuroSys '07, pages 289–302, New York, NY, USA, 2007. ACM.

[122] W. Pan, D. Mu, H. Wu, and L. Yao. Feedback control-based qos guarantees in web application servers. In *International Conference on High Performance Computing and Communications*, pages 328 –334, sept. 2008.

[123] W. Pan, D. Mu, H. Wu, X. Zhang, and L. Yao. Feedback control-based database connection management for proportional delay differentiation-enabled web application servers. In J. Cao, M. Li, M.-Y. Wu, and J. Chen, editors, *Network and Parallel Computing*, volume 5245 of *Lecture Notes in Computer Science*, pages 74–85. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-88140-77.

[124] S. Parekh, N. Gandhi, J. Hellerstein, D. Tilbury, T. Jayram, and J. Bigus. Using control theory to achieve service level objectives in performance management. *Real-Time Syst.*, 23:127–141, July 2002.

[125] S.-M. Park and M. Humphrey. Feedback-controlled resource sharing for predictable escience. In *ACM/IEEE conference on Supercomputing*, SC '08, pages 13:1–13:11, Piscataway, NJ, USA, 2008. IEEE Press.

[126] S.-M. Park and M. Humphrey. Self-tuning virtual machines for predictable escience. In *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, CCGRID '09, pages 356–363, Washington, DC, USA, 2009. IEEE Computer Society.

[127] S.-M. Park and M. Humphrey. Predictable high-performance computing using feedback control and admission control. *IEEE Transactions on Parallel and Distributed Systems*, 22(3):396 –411, march 2011.

[128] T. Patikirikorala, A. Colman, J. Han, and L. Wang. A multi-model framework to implement self-managing control systems for qos management. In *Proceeding of the 6th international symposium on Software engineering for adaptive and self-managing systems*, SEAMS '11, pages 218–227, New York, NY, USA, 2011. ACM.

[129] T. Patikirikorala, L. Wang, and A. Colman. Towards optimal performance and resource management in web systems via model predictive control. In *Australian Control Conference*, AUCC '11, 2011.

[130] T. Patikirikorala, L. Wang, A. Colman, and J. Han. Hammerstein-wiener nonlinear model based predictive control for relative qos performance and resource management of software systems. *Control Engineering Practice*, 20(1):49 – 61, 2011.

[131] C. Poussot-Vassal, M. Tanelli, and M. Lovera. Linear parametrically varying mpc for combined quality of service and energy management in web service systems. In *American Control Conference*, pages 3106 –3111, 30 2010-july 2 2010.

[132] W. Qin and Q. Wang. Feedback performance control for computer systems: an lpv approach. In *American Control Conference*, pages 4760 – 4765 vol. 7, june 2005.

[133] W. Qin and Q. Wang. Using stochastic linear-parameter-varying control for cpu management of internet servers. In *IEEE Conference on Decision and Control*, pages 3824 –3829, dec. 2007.

[134] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu. No "power" struggles: coordinated multi-level power management for the data center. In *Proceedings of the 13th international conference on Architectural support for programming languages and operating systems*, ASPLOS XIII, pages 48–59, New York, NY, USA, 2008. ACM.

[135] A. J. Ramirez and B. H. C. Cheng. Design patterns for developing dynamically adaptive systems. In *Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS '10, pages 49–58, New York, NY, USA, 2010. ACM.

[136] A. Robertsson, B. Wittenmark, M. Kihl, and M. Andersson. Design and evaluation of load control in web server systems. In *American Control Conference*, volume 3, pages 1980 –1985 vol.3, 30 2004-july 2 2004.

[137] M. Salehie and L. Tahvildari. Self-adaptive software: Landscape and research challenges. *ACM Trans. Auton. Adapt. Syst.*, 4:14:1–14:42, May 2009.

[138] E. Sarhan, A. Ghalwash, and M. Khafagy. Specification and implementation of dynamic web site benchmark in telecommunication area. In *Proceedings of the 12th WSEAS international conference on Computers*, pages 863–867, Stevens Point, Wisconsin, USA, 2008. World Scientific and Engineering Academy and Society (WSEAS).

[139] N. Shankaran, X. D. Koutsoukos, D. C. Schmidt, Y. Xue, and C. Lu. Hierarchical control of multiple resources in distributed real-time and embedded systems. In *Proceedings of the 18th Euromicro Conference on Real-Time Systems*, pages 151–160, Washington, DC, USA, 2006. IEEE Computer Society.

[140] P. Shao-Liang, L. Shan-Shan, L. Xiang-Ke, P. Yu-Xing, and Y. Hui. Feedback control with prediction for thread allocation in pipeline architecture web server. In S. Chaudhuri, S. Das, H. Paul, and S. Tirthapura, editors, *Distributed Computing and Networking*, volume 4308 of *Lecture Notes in Computer Science*, pages 454–465. Springer Berlin / Heidelberg, 2006. 10.1007/1194795050.

[141] M. Shaw. Beyond objects: a software design paradigm based on process control. *SIGSOFT Softw. Eng. Notes*, 20:27–38, January 1995.

[142] B. Solomon, D. Ionescu, M. Litoiu, and M. Mihaescu. A real-time adaptive control of autonomic computing environments. In *Proceedings of the 2007 conference of the center for advanced studies on Collaborative research*, CASCON '07, pages 124–136, New York, NY, USA, 2007. ACM.

[143] A. Soria-Lopez, P. Mejia-Alvarez, and J. Cornejo. Feedback scheduling of power-aware soft real-time tasks. In *International Conference on Computer Science*, pages 266 – 273, sept. 2005.

[144] J. Stankovic, T. He, T. Abdelzaher, M. Marley, G. Tao, S. Son, and C. Lu. Feedback control scheduling in distributed real-time systems. In *IEEE Real-Time Systems Symposium, 2001*, pages 59 – 70, dec. 2001.

[145] G. Starnberger, L. Froihofer, and K. M. Goeschka. Abstract only: adaptive run-time performance optimization through scalable client request rate control. *SIGSOFT Softw. Eng. Notes*, 36:39–39, Sept. 2011.

[146] A. J. Storm, C. Garcia-Arellano, S. S. Lightstone, Y. Diao, and M. Surendra. Adaptive self-tuning memory in db2. In *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pages 1081–1092. VLDB Endowment, 2006.

[147] K. J. strom and B. Wittenmark. *Adaptive Control*. Addison-Wesley Publishing Company, 1995.

[148] Y. Tang, X. Luo, Q. Hui, and R. Chang. On generalized low-rate denial-of-quality attack against internet services. In *International Workshop on Quality of Service*, pages 1 –5, july 2009.

[149] A. Tesanovic, M. Amirijoo, K.-M. Bjrk, and J. Hansson. Em-

powering configurable qos management in real-time systems. In *Proceedings of the 4th international conference on Aspect-oriented software development*, AOSD '05, pages 39–50, New York, NY, USA, 2005. ACM.

[150] F. Tian, W. Xu, and J. Sun. Web qos control using fuzzy adaptive pi controller. In *Ninth International Symposium on Distributed Computing and Applications to Business Engineering and Science (DCABES)*, pages 72 –75, aug. 2010.

[151] Y.-C. Tu, S. Liu, S. Prabhakar, and B. Yao. Load shedding in stream databases: a control-based approach. In *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pages 787–798. VLDB Endowment, 2006.

[152] M. Turner, B. Kitchenham, P. Brereton, S. Charters, and D. Budgen. Does the technology acceptance model predict actual use? a systematic literature review. *Information and Software Technology*, 52(5):463 – 479, 2010. ¡ce:title¿TAIC-PART 2008¡/ce:title¿ ¡ce:subtitle¿TAIC-PART 2008¡/ce:subtitle¿.

[153] L. Wang. *Model Predictive Control System Design and Implementation Using MATLAB*. Springer Publishing Company, Incorporated, 2009.

[154] M. Wang, N. Kandasamy, A. Guez, and M. Kam. Distributed cooperative control for adaptive performance management. *IEEE Internet Computing*, 11(1):31 –39, jan.-feb. 2007.

[155] R. Wang, D. M. Kusic, and N. Kandasamy. A distributed control framework for performance management of virtualized computing environments. In *Proceeding of the 7th international conference on Autonomic computing*, ICAC '10, pages 89–98, New York, NY, USA, 2010. ACM.

[156] X. Wang, M. Chen, and X. Fu. Mimo power control for high-density servers in an enclosure. *IEEE Transactions on Parallel and Distributed Systems*, 21(10):1412 –1426, oct. 2010.

[157] X. Wang, M. Chen, C. Lefurgy, and T. Keller. Ship: Scalable hierarchical power control for large-scale data centers. In *International Conference on Parallel Architectures and Compilation Techniques*, pages 91 –100, sept. 2009.

[158] X. Wang, Y. Chen, C. Lu, and X. Koutsoukos. Towards controllable distributed real-time systems with feasible utilization control. *IEEE Transactions on Computers*, 58(8):1095 –1110, aug. 2009.

[159] X. Wang, X. Fu, X. Liu, and Z. Gu. Pauc: Power-aware utilization control in distributed real-time systems. *IEEE Transactions on Industrial Informatics*, 6(3):302 –315, aug. 2010.

[160] X. Wang, D. Jia, C. Lu, and X. Koutsoukos. Deucon: Decentralized end-to-end utilization control for distributed real-time systems. *IEEE Trans. on Parallel and Distributed Systems*, 18:2007, 2007.

[161] X. Wang and Y. Wang. Coordinating power control and performance management for virtualized server clusters. *IEEE Transactions o Parallel and Distributed Systems*, 22(2):245 – 259, feb. 2011.

[162] Y. Wang, R. Deaver, and X. Wang. Virtual batching: Request batching for energ conservation in virtualized servers. In *International Workshop on Quality of Service*, pages 1 –9, june 2010.

[163] Y. Wang and X. Wang. Power optimization with performance assurance for multi-tier applications in virtualized data centers. In *International Conference on Parallel Processing Workshops*, pages 512 –519, sept. 2010.

[164] Y. Wang, X. Wang, M. Chen, and X. Zhu. Partic: Power-aware response time control for virtualized web servers. *IEEE Transactions on Parallel and Distributed Systems*, 22(2):323 –336, feb. 2011.

[165] Z. Wang, Y. Chen, D. Gmach, S. Singhal, B. Watson, W. Rivera, X. Zhu, and C. Hyser. Appraise: application-level performance management in virtualized server environments. *IEEE Transactions on Network and Service Management*, 6(4):240 –254, december 2009.

[166] Z. Wang, X. Liu, A. Zhang, C. Stewart, X. Zhu, and T. Kelly. Autoparam: Automated control of application-level performance in virtualized server environments. In *Workshop on Feedback Control Implementation and Design in Computing Systems and Networks*, 2007.

[167] Z. Wang, C. McCarthy, X. Zhu, P. Ranganathan, and V. Talwar. Feedback control algorithms for power management of servers. In *International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks*, 2008.

[168] Z. Wang, X. Zhu, and S. Singhal. Utilization and slo-based control for dynamic sizing of resource partitions. In *Distributed Systems, Operations and Management*, pages 133–144, 2005.

[169] J. Wei and C.-Z. Xu. Feedback control approaches for quality of service guarantees in web servers. In *American Fuzzy Information Processing Society*, pages 700 – 705, june 2005.

[170] J. Wei, X. Zhou, and C.-Z. Xu. Robust processing rate allocation for proportional slowdown differentiation on internet servers. *IEEE Transactions on Computers*, 54(8):964 – 977, aug. 2005.

[171] L. Wei and H. Yu. Research on a soft real-time scheduling algorithm based on hybrid adaptive control architecture. In *American Control Conference*, volume 5, pages 4022 – 4027 vol.5, june 2003.

[172] M. Woodside, T. Zheng, and M. Litoiu. Service system resource management based on a tracked layered performance model. In *IEEE International Conference on Autonomic Computing*, pages 175 – 184, june 2006.

[173] K. Wu, D. Lilja, and H. Bai. The applicability of adaptive control theory to qos design: limitations and solutions. In *IEEE International Parallel and Distributed Processing Symposium*, page 8 pp., april 2005.

[174] P. Xiong, Z. Wang, S. Malkowski, Q. Wang, D. Jayasinghe, and C. Pu. Economical and robust provisioning of n-tier cloud workloads a multi-level control approach. In *International Conference on Distributed Computing Systems*, pages 571 – 580, june 2011.

[175] C.-Z. Xu, B. Liu, and J. Wei. Model predictive feedback control for qos assurance in webservers. *Computer*, 41(3):66 –72, march 2008.

[176] W. Xu, Z. Tan, A. Fox, and D. Patterson. Regulating workload in j2ee application servers. *International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks (FeBID)*, 2006.

[177] W. Xu, X. Zhu, S. Singhal, and Z. Wang. Predictive control for dynamic resource allocation in enterprise data centers. In *IEEE/IFIP Network Operations and Management Symposium*, pages 115 –126, april 2006.

[178] C. Xu-Dong, Z. Qing-Xin, L. Yong, and X. Guang Ze. End-to-end deadline control for aperiodic tasks in distributed real-time systems. *J. Supercomput.*, 43:225–240, March 2008.

[179] H. Yansu, D. Guanzhong, G. Ang, and P. Wenping. A self-tuning control for web qos. In *International Conference on Information Engineering and Computer Science*, ICIECS, pages 1 –4, dec. 2009.

[180] J. Yao, X. Liu, X. Chen, X. Wang, and J. Li. Online decentralized adaptive optimal controller design of cpu utilization for distributed real-time embedded systems. In *American Control Conference*, pages 283 –288, 30 2010-july 2 2010.

[181] J. Yao, X. Liu, and X. Zhu. Reduced dimension control based

on online recursive principal component analysis. In *American Control Conference*, pages 5713 –5718, june 2009.

[182] C. A. Yfoulis and A. Gounaris. Honoring slas on cloud computing services: a control perspective. In *European Control Conference 2009*, ECC09, 2009.

[183] C. Yu and D. Qionghai. A improved elastic scheduling algorithm based on feedback control theory. In *International Conference on Signal*, volume 2 of *ICSP*, pages 1330 – 1339 vol.2, aug.-4 sept. 2004.

[184] J. Zhang and Y. Zou. Predictive control for performance guarantees in soft real-time scheduling systems. In *The Sixth World Congress on Intelligent Control and Automation*, volume 2, pages 6944 –6948, 0-0 2006.

[185] R. Zhang, C. Lu, T. Abdelzaher, and J. Stankovic. Controlware: a middleware architecture for feedback control of software performance. In *International Conference on Distributed Computing Systems*, pages 301 – 310, 2002.

[186] X. Zhou, Y. Cai, and E. Chow. An integrated approach with feedback control for robust web qos design. *Comput. Commun.*, 29:3158–3169, October 2006.

[187] Y. Zhou and K.-D. Kang. Deadline assignment and tardiness control for real-time data services. In *Euromicro Conference on Real-Time Systems (ECRTS)*, pages 100 –109, july 2010.

[188] X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, P. Padala, and K. Shin. What does control theory bring to systems research? *SIGOPS Oper. Syst. Rev.*, 43:62–69, January 2009.

[189] X. Zhu, Z. Wang, and S. Singhal. Utility-driven workload management using nested control design. In *American Control Conference*, page 6 pp., june 2006.

[190] X. Zhu, D. Young, B. Watson, Z. Wang, J. Rolia, S. Singhal, B. McKee, C. Hyser, D. Gmach, R. Gardner, T. Christian, and L. Cherkasova. 1000 islands: Integrated capacity and workload management for the next generation data center. In *International Conference on Autonomic Computing*, pages 172 –181, june 2008.