An Adaptive Control System for Interactive Virtual Environment Content Delivery to Handheld Devices

Gianluca Paravati, Andrea Sanna, Fabrizio Lamberti, and Luigi Ciminiera

Politecnico di Torino, Dipartimento di Automatica e Informatica,
C.so Duca degli Abruzzi 24, I-10129, Torino, Italy
gianluca.paravati@polito.it,
andrea.sanna@polito.it,
fabrizio.lamberti@polito.it,
luigi.ciminiera@polito.it

Abstract. Wireless communication advances enable emerging video streaming applications to mobile handheld devices. For example, it is possible to show and interact with a complex 3D virtual environment on a "thin" mobile device through remote rendering techniques, where a rendering server is in charge to render 3D data and stream the corresponding image flow to the demanding client. However, due to bandwidth fluctuating characteristics and limited mobile device CPU capabilities, it is extremely challenging to design effective systems for interactive streaming multimedia over wireless networks. This paper presents a novel approach based on a controller able to automatically adjust streaming parameters basing on feedback measures coming back from the client device. Experimental results prove the effectiveness of the proposed solution to cope with bandwidth changes, thus providing a high Quality of Service (QoS) in remote visualizations.

Keywords: Video Streaming to Mobile Devices, Closed-Loop Controller, QoS, Remote Visualization.

1 Introduction

The combination of advances in wireless communication and the rapid evolution and growing popularity of mobile handheld devices can converge into appealing applications within the field of 3D graphics and virtual reality. For instance, virtual guiding malls, multiplayer games and collaborative virtual environments are emerging applications now available also on "thin" devices. The increased capability of wireless connectivity allows to display graphically attractive environments also on devices with limited HW capabilities through remote rendering techniques, where a server is in charge to render 3D data and stream the corresponding image flow to the mobile client [1,2]; the user on the client side can interact with the 3D scene by sending navigation commands to the server.

P. Daras and O. Mayora (Eds.): UCMedia 2009, LNICST 40, pp. 169–178, 2010.

This work presents the design of a novel feedback-based controller in the context of image streaming systems with interactivity constraints; the controller is designed to automatically adapt streaming parameters to both bandwidth fluctuations and device characteristics such as maximum supported resolution and processing capabilities. The proposed technique works at the application level and it can be applied to any kind of client device; however, in this work, we specifically focus on common handheld devices, such as PDAs and smart phones, because of their limited computational capabilities. Preliminary results indicate the effectiveness of the proposed approach, although it is only a starting point for future work investigating control-based remote rendering.

The main features of the proposed system include: increase in visualization quality through smoothly changes in streaming parameter values, adaptation of streaming parameters to bandwidth fluctuations, and adjustment to end device capabilities. The proposed control technique is general and it can be used in any image compression-based streaming scenario.

The remainder of the paper is organized as follows. Section 2 reviews previous works related to the management of streaming parameters and control-theoretic approaches. Section 3 presents the details of the proposed control framework. Experimental results are discussed in Section 4. Finally, conclusions are drawn in Section 5.

2 Related Work

Real-time and interactive streaming to handheld devices over variable bandwidth channels has to deal with different Quality of Service (QoS) issues ranging from network features (such as bandwidth fluctuations and channel latency) to mobile device capabilities (such as receiver decoding performances). The concept of QoS addressed in this paper refers to the issues pertaining the achievement of interactive frame rates and low latencies for interactive remote visualization applications. These issues are investigated in depth in [3]; to deploy effective remote visualization applications, a few QoS requirements need to be met. If these requirements cannot be met, the user will be probably unable to get the expected results. The QoS requirements that are most directly related to remote visualization are: low delay, high throughput/bandwidth, low latency. Motion-JPEG (M-JPEG) has proven to be an effective means for obtaining very low latencies and low processor overhead, although at the expense of an increased bandwidth [4,5]. In fact, M-JPEG parameters like resolution, frame rate and image quality (that determine the bandwidth occupation of the streaming flow) can be combined and tuned to suit network characteristics. The relationship among these parameters can be modeled as in [6]:

$$f = \frac{B}{w \cdot h \cdot C_d \cdot \frac{1}{C_r}}; \tag{1}$$

where f is the achievable frame rate, w and h denote image resolution, B is the currently available bandwidth, C_d is the color depth in terms of bits per pixel, and C_r is the compression ratio of an image with respect to the same uncompressed picture (i.e., C_r is strictly related to the image quality).

Control-theoretic approaches to performance management for computer systems such as Internet web servers, databases and storage systems, have been successfully applied in the past. In [7], a study on the admission control for an Internet web server is presented; a linear-parameter-varying (LPV) approximation for the modeling of the dynamic relationship from the request rejection ratio to the response time for the admitted requests is used. The main characteristic of an LPV controller is the possibility to control state variable non-linear dynamics in different working conditions, due to external agents influencing the system. Although this approach initially seemed to be suitable to control the modification of the above parameters basing on the available bandwidth, the main drawback that prevents from using it in a M-JPEG streaming scenario is due to the image quality dynamics; indeed, its impact on the image size to be delivered (and hence on the bandwidth occupation) is unpredictable.

Since working conditions in interactive streaming to mobile handheld devices can differ with available channel bandwidth, the attention was focused on adaptive control techniques [8,9], in order to build a controller able to modify its control parameters according to state variations. Adaptive control techniques estimate the behavior of the system through linear regression algorithms, which functionality is based on the refinement at each step to asymptotically reach a set of parameters able to represent the unknown initial system. In video streaming context, these methods are not suitable because of the large fluctuations of the parameters; an estimator based on these techniques cannot identify an asymptotically stable system.

After establishing that the application of complicated control techniques does not favor the solution of the problem, it has been preferred to design a PID-based controller; its robustness and reliability are particularly useful to control a system characterized by unpredictable fluctuations and to correctly exploit the feedback channel of the system. In order to achieve performance specifications, gain-scheduling techniques were used. Gain scheduling [10] is based on the idea of using different (a-priori performed) calibrations in different circumstances, thus realizing a parameter calibration system able to adapt to the state of the system.

3 The Control Algorithm

The design of the proposed controller aims at taking advantage of automatic control techniques to concurrently tune all the parameters involved in a remote rendering scenario (i.e. resolution, image quality, frame rate) without any a-priori knowledge about the precise effects caused by altering these parameters. The main requirements identified in the design phase of the controller are: sensitivity to feedback measures, robustness to non linearity, independence to the quality of the underlying network, and optimal usage of available resources.

The control system can be sketched as in Figure 1. The controller is fully implemented on the server side, meaning that the client has only to compute its

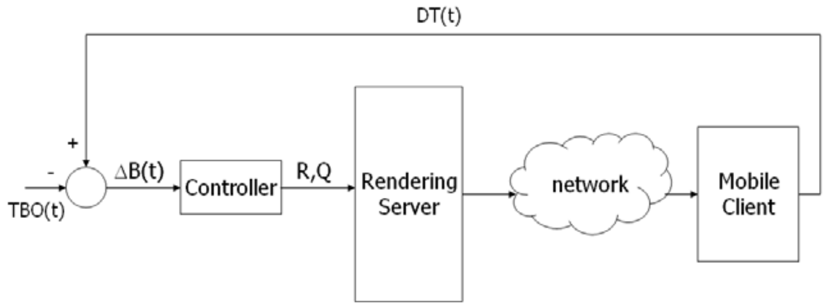

Fig. 1. The logical control system

own device throughput and to periodically feed this information to the streaming server. It is worth remarking that, in this paper, the term robustness is related to the degree to which the controller can function correctly in the presence of uncertainties affecting our model of the system.

The reference of the control system is the theoretical bandwidth occupation $TBO(t)$ that depends on the current encoding parameters and can be computed reversing eq. 1. The measured output of the system (on the client side) is the device throughput $DT(t)$. The general structure of the controller system is characterized by a single input (the bandwidth error $\Delta B(t) = DT(t) - TBO(t)$) and two separate outputs: a resolution (R) and a quality (Q) factor. These outputs are used by the rendering server to create, compress, and stream through the network an image representation of the 3D scene to the mobile device.

The controller deals with the bandwidth control trying to balance the system around a target frame rate (default or supplied by the user); the value of the target frame rate influences $TBO(t)$. Basically, the actual frame rate of the client (computed using eq. 1) is continuously compared to the target frame rate; the controller tries to minimize the error/difference between these two values; the system variables are controlled so as to undergo changes that are proportional to the error measured as input. In more details, the system changes the parameters of the encoded flow of images in such a way to reach the desired frame rate exploiting the minimum between the current available bandwidth (that depends on network status) and the current device throughput (that depends on device capabilities).

The reciprocal interaction between the controlled variables (R and Q) causes non linear and unpredictable fluctuations in the required bandwidth to be used for the transmission of the data stream to the client. This issue can be tackled through the isolation of the relationship that binds each variable to the image size, so that we can establish the degree of variation needed to increase or decrease the image size (and so the bandwidth occupation) to a determined quantity.

The basic idea is to subdivide the available bandwidth in different portions to be assigned to each system variable; thus, the input bandwidth error ΔB

can be splitted into two separated quantities: ΔB_R and ΔB_Q. The knowledge of mathematical relations able to govern the growth or the reduction of a system variable, as the bandwidth changes, allows to individually set each variable by keeping fixed all the other parameters. A mathematical relationship that links a bandwidth quantity to be filled (or released) and a resolution increment (or decrement) has been defined:

$$\Delta R = \frac{\Delta B}{B_R^{up} - B_R}; \tag{2}$$

where ΔR is the amount of the increment or decrement in resolution, ΔB is the bandwidth error measured by the controller, B_R and B_R^{up} are respectively the bandwidth to be used with the current resolution index and the bandwidth to be used with the immediately upper resolution index. The proportionality between resolution and bandwidth is clearly visible from eq. 2; on the other hand, eq. 1 makes explicit the relationship between frame rate and bandwidth, showing how a resolution change can be reflected on a frame rate change through a proportionality coefficient k_R:

$$\Delta R = k_R \cdot \Delta f. \tag{3}$$

The controller deals with another system variable represented by the compression quality. The identification process of a control relationship for this variable has been more complicated, mainly because it is difficult to evaluate, even approximately, a mathematical relationship between image quality and compression factor, which changes reflect to the compressed image size through an inverse proportionality relationship. Thereby, it has been decided to exploit the controller feedback measure, slowly varying the quality level at each loop and then correcting the excessive bandwidth repercussions if necessary. The idea is to subsequently control a variable that cannot be controlled a priori because of its unpredictability. Since a precise bandwidth variation implies a precise frame rate variation, a direct relationship between image quality and frame rate is established. Actually, these variables are not linearly dependent because the incidence on the frame rate is determined by the compression factor C_r, that has a non-deterministic dependence on image quality. Such a relationship can be modelled as:

$$\Delta Q = k_Q \cdot \Delta f. \tag{4}$$

The instability both of the available bandwidth and of the device throughput makes unstable the input reference error ΔB of the controller; thus, it becomes vacillating and it does not asymptotically approach to a stable equilibrium value. Every instant can be characterized by different working conditions. In order to manage this instability an adaptive control approach can be used. It allows to find some specific coefficients for the control equations; these coefficients are relative to external system variables and the current state of the system. The k_R coefficient can be expressed as:

$$k_R = \frac{k_p}{k_B} = \frac{k_p}{B_R^{up} - B_R}; \tag{5}$$

where k_p is a proportional constant of a PID-based system, k_B is an adaptive parameter that depends on the current resolution. In this way, k_R is constituted by a part used to compute the bandwidth occupied by a change in resolution, represented by k_p, and a part used by the system to tune the bandwidth quantity to be assigned to the resolution. As a result, the input bandwidth is subdivided into a portion that is assigned to the resolution variable. As the coefficient k_p increases a greater portion of bandwidth is reserved to allow a change in resolution, enhancing the sensitivity of the system to bandwidth fluctuations and the quick response to reference tracking as a matter of fact. The unpredictable nature of bandwidth fluctuations, particularly in wireless connections, and the precision that is possible to reach estimating the bandwidth as a function of the resolution index, lead the integral-derivative part of a PID-based controller useless at the moment.

The subdivision of the bandwidth between the system variables is a key aspect for the controller (calibration phase). Indeed, the two variables (resolution and image quality) cannot be precisely tuned at the same time because of their reciprocal interactions; on the other hand, it is possible to act singularly on each of them through the relationships between these variables and the reference parameter, i.e. the target frame rate. Thus, the control phase has been split in two stages. In every stage, it is possible to act on a single variable by proceeding in cascade, having a different bandwidth as an input at each stage and keeping fixed all the parameters to modify only the considered variable. At each stage the input bandwidth is used to modify the variable of interest and it progressively decreases till exhausting, thus minimizing the input error as a matter of fact. In this way the system can be characterized as a SISO model. A first stage receives as input the bandwidth error ΔB and uses part of it to regulate resolution (according to eq. 2); the new resolution value causes a change in theoretical bandwidth occupation and the achievable frame rate computed with the current parameters, thus it is used as input of a second stage to recompute the bandwidth error. The second stage receives as input the new bandwidth error that is used to regulate quality according to eq. 4.

4 Tests and Results

The proposed controller has been implemented and tested in a remote rendering scenario. The rendering server runs on a Dual-Core AMD Opteron CPU 2.60 GHz workstation equipped with 3.50 GB of RAM and with an NVIDIA Quadro FX 3500 graphics card; it has been developed using the C++ language and it is based on the OpenSG library. The client program runs on a HTC TyTN II smart phone, connected to the rendering server through a 802.11g wireless access point. The client program has been developed using J2ME (Java Micro Edition), as it provides a flexible environment for applications running on mobile and other embedded devices. Figure 2 shows a remote rendering session; the server (shown in the background) is in charge of rendering the 3D scene and of streaming a M-JPEG flow to the “thin” client (shown in the foreground).

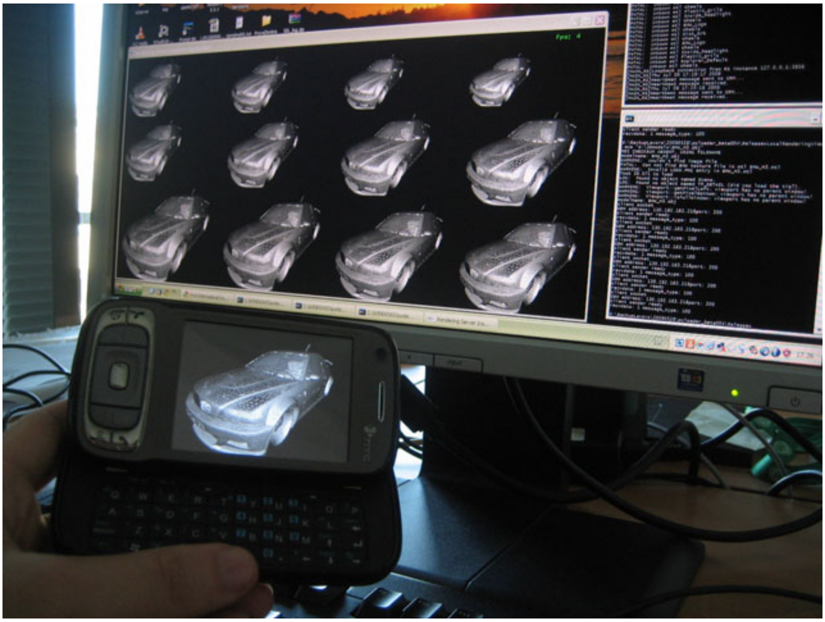

Fig. 2. A remote 3D rendering session on a TyTN II smart phone

Figure 3 shows the behavior of the controller under varying system conditions; the monitored parameters are (from top to bottom): resolution, image quality, actual frame rate reached by the mobile device, and measured throughput (the feedback feature). Figure 3 (a) shows the ability of the controller to react to changes in network conditions. The system is configured to maintain a target frame rate (FR_{target} = 15 fps) during the whole test. Indeed, the client application allows the user to express his/her preference in terms of frame rate; a higher frame rate enhances motion smoothness and impacts on perceived interactivity. During the first part of the simulation (period between $t = 0$ and $t = t_1$) the system worked in stationary conditions exploiting the maximum throughput of the device ($DT \approx 60$ KB/s); a trade-off between resolution and quality was reached allowing to maintain the target frame rate. At time $t = t_1$ a maximum bandwidth limitation was imposed on the server side ($BW_{limit} = 40$ KB/s) to simulate a network bottleneck; the frame rate on the client side rapidly dropped down (from $FR \approx 15$ fps to $FR \approx 9$ fps) and the controller reacted by gradually reducing the parameter values. While changing parameters, the controller continuously compared the target frame rate with the effective frame rate. After the $t = t_2$, the controller leaded the system to a different steady state reaching again the target frame rate, thus mitigating the effects of the bandwidth bottleneck previously introduced. If the target frame rate was higher, parameters would have continued to be reduced until the target frame rate was reached again. Figure 3 (b) proves the ability of the controller to follow a specified target

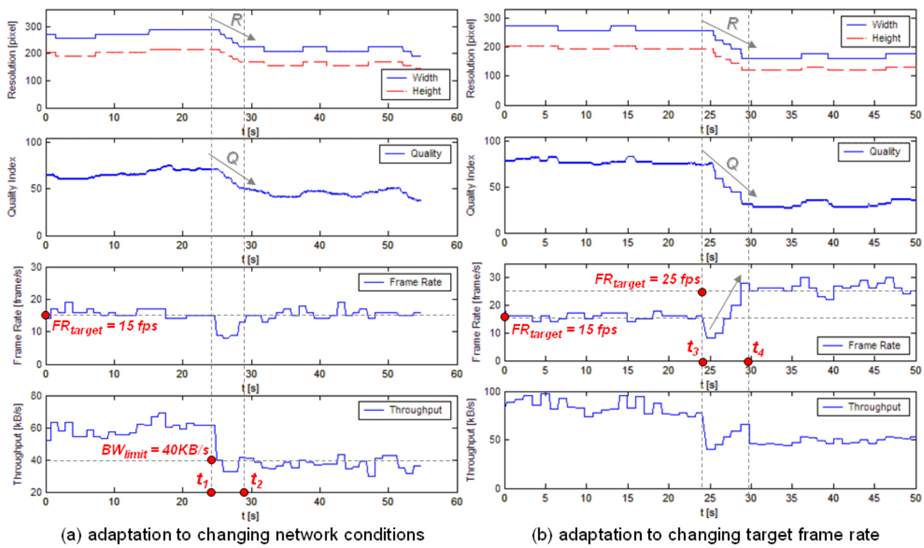

Fig. 3. Performance on a TyTN II smart phone: (a) adaptation to changing network conditions and (b) adaptation to changing target frame rate

frame rate. Initially, the system maintained a target frame rate equal to 15 fps (period between $t = 0$ and $t = t_3$); the system was steady around high values of resolution ($R = 288 \times 216$ pixels) and quality ($Q \approx 80$) parameters. At time $t = t_3$ the target frame rate was set to 25 fps through the client GUI. During the period between $t = t_3$ and $t = t_4$ the controller reacted by gradually reducing resolution ($R = 176 \times 132$ pixels) and quality ($Q \approx 28$). After the transitory, the system reached a different steady state around the new frame rate value; although the growth of the number of frames per second received, the device throughput is lower than before because images are more compressed, thus indicating that performances are limited by the decoding capabilities of the mobile device.

In this work, a set of experimental tests of user experience were carried out in order to evaluate its relationship with QoS parameters. A group of 56 subjects was asked to carry out two sets of tests on the designed system and later answer a questionnaire aimed at collecting a feedback on system performances. Each user was individually trained in order to know how to use features of the remote visualization application running on an HTC TyTN II mobile device. The trainer performed two sessions (with and without the proposed controller) showing the aspect and behavior of the user interface; in particular, the possibility of changing the target frame rate was emphasized. After the training phase, each user was allowed to use the device alone. Each user was asked to perform the two tests by navigating the 3D scene; during the first test the controller was disabled, while during the second test it was enabled. During the first test, the image stream

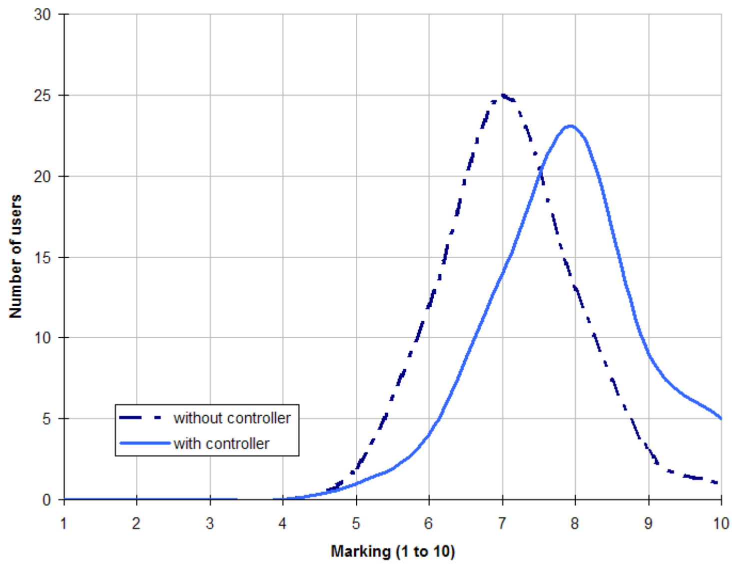

Fig. 4. User feedbacks from user experience tests

characteristics were fixed; in particular, encoding parameters were configured to match the maximum device resolution with the best image quality. This choice led to a poor frame rate. During the second test, the controller was enabled and users were allowed to change the target frame rate. In the questionnaires, users were then asked to assign marks (in a range 1-10) to the perceived performance of the visualization tool both with and without the controller. Figure 4 shows the results of this subjective evaluation. The x-axis represents the possible mark values, while the y-axis indicates the number of users assigning the given mark. The broken line is related to the score distribution obtained by experimenting the visualization system without the proposed controller. On the other hand, the unbroken line refers to the score distribution achieved by enabling the control system. Figure 4 clearly shows that application of the control system presented in this paper noticeably improved the user experience, that is the perceived performance of the visualization. The average mark obtained using the described methodology is equal to 7.89, compared to 7.1 when it is not used. This can constitute an effective proof of the relationship between the user experience and QoS parameters.

5 Conclusions

Due to bandwidth-demanding characteristics and limited mobile device capabilities, it is extremely challenging to design effective systems for real-time and interactive streaming multimedia over wireless networks for handheld devices. This paper presents a controller able to automatically adjust M-JPEG streaming parameters; the solution has been implemented and tested in a remote rendering scenario, where smart phones are able to interactively display complex 3D virtual environments although their limited computational capabilities. However, the proposed solution is general and can be applied to any interactive streaming scenario; moreover, this control technique can be applied to video compression schemes such as MPEG. The controller is implemented on the server side; it exploits a feedback measure from clients thus it is able to continuously adapt to changing network conditions, different device throughput capabilities and different interaction requirements.

References

1. Pazzi, R.W.N., Boukerche, A., Huang, T.: Implementation, Measurement, and Analysis of an Image-Based Virtual Environment Streaming Protocol for Wireless Mobile Devices. IEEE Transactions on Instrumentation and Measurement 57(9), 1894–1907 (2008)
2. Quax, P., Geuns, B., Jehaes, T., Lamotte, W., Vansichem, G.: On the Applicability of Remote Rendering of Networked Virtual Environments on Mobile Devices. In: International Conference on Systems and Networks Communications, p. 16 (2006)
3. Stegmaier, S., Diepstraten, J., Weiler, M., Ertl, T.: Widening the Remote Visualization Bottleneck. In: Proc. Third Int'l Symp. Image and Signal Processing and Analysis, pp. 174–179 (2003)
4. Endoh, K., Yoshida, K., Yakoh, T.: Low delay live video streaming system for interactive use. In: IEEE International Conference on Industrial Informatics, pp. 1481–1486 (2008)
5. Nishantha, D., Hayashida, Y., Hayashi, T.: Application level rate adaptive motion-JPEG transmission for medical collaboration systems. In: The 24th International Conference on Distributed Computing Systems Workshops, pp. 64–69 (2004)
6. Paravati, G., Sanna, A., Lamberti, F., Ciminiera, L.: A novel approach to support quality of experience in remote visualization on mobile devices. In: Eurographics 2008 short paper Proceedings, pp. 223–226 (2008)
7. Qin, W., Wang, Q.: An LPV approximation for admission control of an internet web server: Identification and control. Control Engineering Practice 15, 1457–1467 (2007)
8. Lu, Y., Abdelzaher, T., Lu, C., Tao, G.: An adaptive control framework for QoS guarantees and its application to differentiated caching services. In: The 10th IEEE International Workshop on Quality of Service, pp. 23–32 (2002)
9. Wu, K., Liljia, D.J., Bai, H.: The applicability of adaptive control theory to QoS design: limitations and solutions. In: The 19th IEEE International Parallel and Distributed Processing Symposium (2005)
10. Slotine, J.J., Li, W.: Applied Nonlinear Control. Prentice-Hall, Englewood Cliffs (1991)