УДК 681.51(045)

M. Aiman Al Akkad, PhD in Engineering, Kalashnikov Izhevsk State Technical University

DELICATE CONTROL VERSUS ROBUST CONTROL

This paper focuses on the need for a multifunction control to be used in robot systems and automation. First it refers to fine motion in the dexterous space and introduces delicate control, then discusses robustness and adaptive control to be used for coarse motion in the work space. Coordination between multiple controllers and scheduling between tasks were considered, and a reconfigurable structure was suggested. Finally it is concluded that a dual control structure with a concurrent elevator scheduling algorithm is sufficiently efficient for a wide range of applications.

Key words: delicacy, robust control, adaptive control, reconfigurable systems, fault tolerance.

Nowadays industrial production is facing increasing competition and high requirements in terms of quality and customer satisfaction. Numerous efforts were spent on analysis and control of hybrid, switched, uncertain systems, and non-switched dynamical systems, but still there is not any method or technique available for control which guaranties good performance and stability. Sufficient conditions for stability such as Lyapunov functions and average dwell time are the most commonly studied, in condition that all subsystems share a common Lyapunov function [4].

The corresponding controller design requires gains to be adjusted to guarantee the stability of each configuration and knowledge of worst case decay rates among subsystems in order to compute the maximum admissible switching speed. Dealing with uncertainty is also of interest so that switching control has been introduced to deal with uncertain time-invariant systems, known as multiple model adaptive or supervisory control where it is switched between candidate controllers based on some estimation to control the system [6].

The supervisory control uses linear time invariant controllers and estimation based switching between controllers is made in order to implement the best possible controllers. The safe multiple model control requires that any change in a controller is small enough so that it does not result in an unstable closed loop and the initial controller chosen is a stabilizing controller.

Methods based on conventional adaptive control architecture have better stability potential for controller switching via reinitializing the adaptation by switching between fixed estimates or resetting the adaptive estimate during transients. However, improving transients is possible only if such estimates are good [3].

The robustness with respect to disturbances and parameter variations should be considered, and adaptive control is used to deal with system uncertainty. The problem with conventional adaptive controllers is that the transient performance is not characterized and stability with respect to bounded parameter variations or disturbances is not guaranteed.

Robust adaptive controllers were developed to address the presence of disturbances and non-parametric uncertainties, based on adaptation laws that require a priori known bounds on parameters, and disturbances, in order to ensure state bound [7]. Extensions to some classes of time varying systems developed is restricted to smoothly varying parameters with known bounds and typically require additional restrictive conditions such as slowly varying unknown parameters or constant and known input vector parameters. Practically errors should be reduced by increasing the adaptation or feedback gains or using a better nominal estimate of the system parameters. The performance with respect to rejection of disturbances as well as transient response needs further investigation. This provides a strong stability and performance robustness for time varying switched systems [2, 8].

Despite the simplicity of formulating this type of problem, it must be noted that to date there is no one method able to solve all possible scenarios. Sensors were used to build a model of the environment, an analytical engine planned a course, and the resulting plan was used to drive the robot. Unfortunately, these robots tended to be slow and unreliable. Building a world model was computationally expensive and could get outdated as the plan was being constructed.

Delicacy and Reactive Control

Delicate control is the control of output to a precision on the order of the precision of the input. Early attempts at mobile robotics were dominated by model-building approaches using the sense, plan, and act architecture. By means of sensors and encoders the required parameters are defined, and a simple proportional control law is used to control the robot. The pendulum balancing robot is an example of such problems.

Delicacy where extreme precision is required can be met in the field of medical surgery. Reactive control demonstrated exceptional speed and robustness yet required very limited computational power. Instead of building a model, robots react to their environment, with higher-level behaviors overriding lower-level ones as necessary to achieve more complex behavior.

Though inadequate for some tasks, responding directly to sensors can be enough to achieve basic obstacle avoidance or wall-following behavior. Ambidextrous robotic arms when performing fine motor tasks it is more efficient to use a delicate reactive control [9].

Robustness and Adaptive Control

Robust control is to find a good set of constant gains despite the motion of the poles, guaranteed to remain in

Получено 05.07.13

favorable locations, or variable gains are precomputed which change with robot configuration so that the system's quasi-static poles remain in fixed positions. In the model-based control, parameters of the manipulator are not known exactly. When the parameters in the model do not match the parameters of the real device, servo errors will result. These servo errors could be used to drive some adaptation scheme that attempts to update the values of the model parameters until the errors disappear.

Adaptive control differs in that it does not need a priori information about the bounds on the uncertain or time-varying parameters; robust control guarantees that if the changes are within given bounds the control law need not be changed, while adaptive control is concerned with control laws changing themselves. The foundation of adaptive control is parameter estimation, where estimation include recursive least squares and gradient descent. Both of these methods provide update laws which are used to modify estimates in real time.

Lyapunov stability is used to derive these update laws and show convergence criterion. Normalization is used to improve the robustness of estimation algorithms. Most simple robot controllers do not use a model-based component at all in their control law, where each joint is controlled as a separate control system. The model-based controller control law can't be implemented on a strict joint-by-joint basis, the controller architecture must allow communication between the joint controllers or must make use of a central processor rather than individual-joint processors. There are various ways to simplify the dynamic equations of a particular manipulator. After the simplification, an approximate decoupling and linearizing law can be derived.

The use of feed forward control has been proposed as a method of using a nonlinear dynamic model in a control law without the need for complex and time-consuming computations to be performed at servo rates. The model-based control is outside the servo loop, so it is possible to have a fast inner servo loop, consisting simply of multiplying errors by gains, with the model-based torques added at a slower rate. In this architecture, the dynamic parameters can be updated at a rate slower than the rate of the closed-loop servo. Manipulator dynamic model is often not known accurately, friction model, ageing, and the tool, change the dynamics of the manipulator.

Probably the most important example of a robust control technique is which minimizes the sensitivity of a system over its frequency spectrum, and this guarantees that the system will not greatly deviate from expected trajectories when disturbances enter the system. An emerging area of robust control from application point of view is sliding mode control which is a variation of variable structure control. Other robust techniques include quantitative feedback theory, and gain scheduling [5].

Coordinator and Scheduling

Including several kinds of control functions in one device is necessary for dynamic tasks held by robots. Movements in certain moment are simple and coarse, in another, are sophisticated and fine. Offering efficient and powerful solutions to scheduling problems constitutes an important challenge. The coordinator role is to give command to the appropriate controller in the right moment where scheduling the tasks for each controller is based on the concurrent elevator algorithm.

In our robotic ranger system two control modes were made for the vehicle motors, where in rough terrains the speed of the motors is gradually decreased and control shifts to the climbing mode [1]. While in our ambidextrous robot system mentioned in our paper [10], two control modes were used for the two ambidextrous robot arms to coordinate the movement between rough ones in the 3D work space and fine ones in the dexterous space.

A promising solution to behavior coordination is switching control, which means there is one and only one controller active at any given time. Figure 1 shows the suggested system structure. However, switching between different controllers might cause unstable behavior even if each individual controller is stable. As an extension of Lyapunov stability theory, multiple Lyapunov function is recommended as a tool to determine the stability of switched systems and hybrid systems.

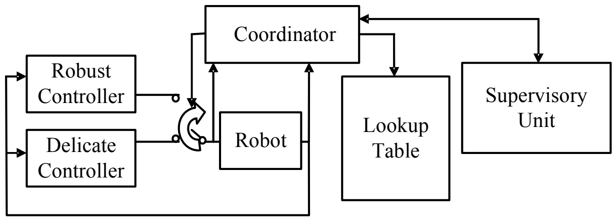

Figure 1. Switching Controller Architecture

The coordinator chooses between the two controllers by consulting a lookup table. Each of these controllers has its own control law and is assigned to the appropriate task. A connected computer calculates the necessary parameters off-line, defines the tasks, and send the necessary information to the lookup table. Tasks are scanned upwards in reference to the current task and the nearest is served, then the process is reversed downwards. Figure 2. Shows the look-up table structure.

Controller ID	Task/Subtask parameters per System Cycle
ID_d	$\theta_{d1}, \dot{\theta}_{d1}, \ddot{\theta}_{d1} \dots \theta_{dn}, \dot{\theta}_{dn}, \ddot{\theta}_{dn}$
ID_r	$\theta_{r1}, \dot{\theta}_{r1}, \ddot{\theta}_{r1} \dots \theta_{rn}, \dot{\theta}_{rn}, \ddot{\theta}_{rn}$

Figure 2. Structure of the look-up table

A. Algorithm: Consists of main program which creates two concurrent processes each executes the corresponding controller task (delicate or robust), where executing the tasks within each of the controllers is performed based on the elevator algorithm.

```
main ()
{ int ID1; int ID2;
ID1 = fork();
if (ID1 == 0) && (Movement==coarse)
  { while (1)
    { Signal=ID1; turn=0;
  }
ID2 = fork ();
```

```
if (ID2 == 0) && (Movement=fine)
  { while (1)
    { Signal=ID2; turn=1;
  }
```

In the task code *controller()* is a function which chooses the appropriate controller and the deliver him the task through the two dimension array *task* which has two fields a pointer to the current controller) and a vector of the corresponding tasks.

```
if  i < N {
  while i <= N
    { controller(Signal, task[Signal,i]);
      Direction=up;
    }
  }
else
  { while i>= 0
    { controller (Signal, task[Signal,i]);
      Direction=down;
    }
  }
```

The concurrent controller processes have a mutual exclusion strategy:

while (TRUE){
 while(turn != 0)
  critical_region();
 turn = 1;
noncritical_region();
}
	while (TRUE) {
	 while(turn != 1)
	  critical_region();
	 turn = 0;
	oncritical_region();
	}

The dynamic model is computed as a function of the desired path only, so when the desired path is known in advance, values could be computed off-line in the supervisory unit before motion begins. At run time, the precomputed torque histories would then be read out of memory.

Likewise, if time varying gains are computed, they too could be computed beforehand and stored. Hence, such a scheme could be quite inexpensive computationally at run time and thus achieve a high servo rate. The dynamic model expressed in its configuration space so that the dynamic parameters of the manipulator will appear as functions of manipulator position only, which are computed by a background process in the supervisory unit.

Conclusion

Different approaches were used in achieving good control, each proved its efficiency in certain applications and under certain constraints, like delicate, reactive, robust, and adaptive.

Having a solution where several controllers are involved in the task offers a more quick and flexible handling in controlling the robot or robots performing the required movements and reach a more practical and precise execution.

Our experiments showed that using dual controller device with a coordinator and a lookup table offers a more generic, speedy, and precise solution for a wide range of robotic applications.

References

1. Designing and Implementing a Robotic Ranger to Scan Rough Terrains / M. A. Al Akkad et. al. // ISTU bulletin. – 2012. – No. 1.
2. Dynamics and Robust Control of Robot-Environment Interaction / Y. Ekalo, et. al. – St. Petersburg Electro-technical University, 2009.
3. *Shi J.*, *Zhou S.* Quality control and improvement for multistage systems : A survey // IIE. – 2009.
4. *Pires J. N.* Industrial Robots Programming: Building Applications for the Factories of the Future. – Springer, 2007.
5. *Al Rifai K.* Flexibility and Robustness in Scheduling. – MIT, 2007.
6. *Calafiore G.*, *Dabbene F.* Probabilistic and Randomized Methods for Design under Uncertainty. – Springer Verlag London Ltd., 2006.
7. Feedback control of dynamic systems / Franklin et al. – Prentice Hall, 2005.
8. *Paoli A.* Fault Detection and Fault Tolerant Control for Distributed Systems. – University of Bologna, 2004.
9. *Al Akkad M. A.* Exploiting two ambidextrous robot arms in achieving cooperative tasks // to be published in ISTU bulletin, 2013.
10. *Al Akkad M. A.* Robotics as an efficient tool in education // EQ– 2012 conference, ISTU, 2012.

Аль Аккад М. Айман, кандидат технических наук, Ижевский государственный технический университет имени М. Т. Калашникова

Высокоточное управление относительно робастного управления

Данная статья посвящена необходимости многофункционального управления в робототехнических системах и системах автоматизации. Прежде всего описывается мелкая моторика в правостороннем пространстве и вводится понятие высокоточного управления, а затем обсуждается надежность и применение адаптивного управления для грубой моторики в рабочем пространстве. Рассматривается взаимодействие между несколькими управляющими устройствами и планирование задач, предлагается структура с перестраиваемой конфигурацией. В заключение делается вывод о том, что двойственная структура управления, использующая параллельный алгоритм лифта для планирования движения, является достаточно эффективной в широком диапазоне областей применения.

Ключевые слова: точность, робастное управление, адаптивное управление, система с перестраиваемой конфигурацией, устойчивость к ошибкам.