

Министерство образования и науки Российской Федерации
Уральский федеральный университет
им. первого Президента России Б. Н. Ельцина

Я. Л. Либерман, К. Ю. Летнев

**Автоматическое устранение
колебаний груза на гибкой подвеске
при подъеме и опускании**

Екатеринбург
УрФУ
2012

УДК 62.752.6: 621.86

ББК 39.9 – 04я73

Л65

Рецензенты:

профессор, доктор технических наук В. В. Каржавин (Российский государственный профессионально-педагогический университет);

действительный член Международной академии авторов научных открытий и изобретений, Заслуженный изобретатель РСФСР Л. А. Каргаев

Либерман, Я. Л.

Л65 Автоматическое устранение колебаний груза на гибкой подвеске при подъеме и опускании : монография / Я. Л. Либерман, К. Ю. Летнев. – Екатеринбург : УрФУ, 2012. – 215 с.

ISBN 978-5-321-02103-3

В книге рассматриваются вертикальные колебания груза на гибкой подвеске как одна из существенных причин аварийности и травматизма при эксплуатации грузоподъемных кранов. Описываются устройства, предназначенные для устранения таких колебаний. Приведены исследования колебаний и математико-технические средства их выполнения. Предлагаются новые устройства устранения колебаний, оценивается их экономическая эффективность.

Библиогр. : 32 назв. Рис. 63. Табл. 9.

УДК 62.752.6: 621.86

ББК 39.9 – 04я73

ISBN 978-5-321-02103-3

© Уральский федеральный
университет, 2012

© Либерман Я. Л.,
Летнев К. Ю., 2012

ОГЛАВЛЕНИЕ

Введение	5
1. Аварии и предаварийные ситуации при эксплуатации грузоподъемных машин и их причины	7
1.1. Статистический анализ травматизма и аварийности при эксплуатации грузоподъемных кранов и иных подъемных сооружений.....	7
1.2. Причины аварий и предаварийных состояний грузоподъемных кранов и колебания груза при подъеме и опускании как особо значимые из них	15
2. Современные автоматические средства устранения вертикальных колебаний грузов, перемещаемых краном	32
2.1. Устройства активного виброгашения с измерением частоты вынужденных колебаний	32
2.2. Системы двухступенчатого торможения с измерением полупериода колебаний	45
2.3. Устройства многоступенчатого и бесступенчатого управления приводами подъема–опускания грузов.....	63
3. Исследование вертикальных колебаний груза на гибкой подвеске при торможении	82
3.1. Задачи и методика исследования.....	82
3.2. Экспериментальная установка	83
3.3. Проведение экспериментов и порядок фиксации их результатов ...	94
3.4. Обработка экспериментальных данных и построение математических моделей	101
4. Модернизация существующих средств устранения вертикальных колебаний грузов на основе проведенного исследования	135
4.1. Модернизация магнитореологического виброгасителя	135

4.2. Модернизация виброгасителя с автоматическим управлением статической осадкой груза на мембране	139
4.3. Модернизация системы двухступенчатого торможения с определением полупериода вынужденных колебаний на основе измерения длины каната и массы груза, перемещаемого краном	141
4.4. Компьютерная система плавного разгона–торможения привода подъема–опускания груза, перемещаемого краном	148
5. Оценка экономической эффективности средств устранения колебаний грузов и снижения аварийности и травматизма при эксплуатации кранов	163
Библиографический список	173
Приложение	177

ВВЕДЕНИЕ

Развитие промышленного производства, строительства и других видов технической деятельности человека сопровождается его естественным стремлением сделать свой труд более производительным, качественным и легким. Это привело к интенсификации создания разнообразных средств механизации и автоматизации, число которых в мире с каждым годом увеличивается. При этом растет и сложность этих средств. В таких условиях все острее становится вопрос обеспечения надежности и безопасности машин и механизмов, устройств управления ими, а также их отдельных узлов и элементов. Пренебрежительное отношение к этому вопросу обходится человеку чрезвычайно дорого. Об этом свидетельствуют участвовавшие техногенные катастрофы, аварии и производственный травматизм, нередко со смертельным исходом. Разумеется, разные технические объекты должны удовлетворять и разным требованиям к их надежности и безопасности. Вряд ли сопоставимы требования такого рода, например, к шариковой авторучке и автомобилю. Транспортные средства, различные подъемные сооружения и им подобные должны отвечать особым требованиям. Они должны быть особенно надежными и безопасными, для чего их зачастую снабжают специальными, обеспечивающими безопасность приборами и приспособлениями. В настоящее время в распоряжении проектировщиков имеется широкий спектр принципов построения подобных приспособлений, типовых конструкций и методов их рационального использования. Ориентировано все это, как правило, на тот или иной вид технических объектов. Не являются исключением и грузоподъемные машины, для которых также создаются специальные устройства повышения их надежности и безопасности. Об одном из видов таких устройств, предназначенных для указанных машин, и идет речь в предлагаемой работе. В ней анализируется аварийность грузоподъемных машин, ее причины, рассматриваются известные устройства для ее снижения и предлагаются их усовершенствованные варианты. Работа по-

священа лишь одному направлению повышения надежности и безопасности грузоподъемных машин, однако оно, по мнению авторов, является одним из наиболее результативных, а потому заслуживает серьезного внимания. Учитывая это, авторы полагают, что данная работа может быть полезна как сложившимся специалистам, так и студентам, обучающимся по специальностям, связанным с проектированием и эксплуатацией соответствующей техники.

Пользуясь случаем, благодарим В. А. Баранова, предоставившего нам материалы для гл. 1 книги и частично для гл. 3 [1]. Благодарим также редактора книги, вложившего немалый труд в подготовку книги к изданию.

1. АВАРИИ И ПРЕДАВАРИЙНЫЕ СИТУАЦИИ ПРИ ЭКСПЛУАТАЦИИ ГРУЗОПОДЪЕМНЫХ МАШИН И ИХ ПРИЧИНЫ

1.1. Статистический анализ травматизма и аварийности при эксплуатации грузоподъемных кранов и иных подъемных сооружений

Подъемные сооружения широко используются во всех отраслях промышленности и сельского хозяйства. Наиболее сложным и распространенным их представителем является грузоподъемный кран. Он представляет собой универсальную грузоподъемную машину циклического действия с механическим приводом, управляемую машинистом (оператором) крана и предназначенную для подъема и последующего перемещения по сложной пространственной трассе различных грузов с помощью индивидуально управляемых механизмов, расположенных на несущей конструкции.

Грузоподъемный кран изначально является устройством повышенной опасности, поскольку при его эксплуатации рабочая зона некоторого рабочего места является и опасной зоной этого места, представляющей профессиональную опасность для машиниста крана и обслуживающего персонала и опасность вообще для посторонних лиц. Опасные и вредные производственные факторы, которым в этих условиях подвергаются люди, связаны в основном с работой на высоте, непреднамеренным контактом с движущимися частями оборудования, возможным ударом от падающих грузов при обрыве стропа, а также авариями самого кранового оборудования и др.

Особенностью грузоподъемного крана по сравнению с другими видами производственного оборудования является то, что его опасная зона подвижна, зависит от выполнения тех или иных погрузочно-разгрузочных, транспортных и складских работ. И это характерно для кранов любых типов: мостовых, кабельных и стреловых.

В кранах мостового типа грузозахватный орган подвешивается к тележке, перемещающейся по пролетному строению в виде моста. Разновидностями кранов мостового типа являются собственно мостовые краны, краны козловые и полукозловые.

В кранах кабельного типа тележка с грузозахватным органом перемещается по несущему канату, зачастую на значительные расстояния (500 и более метров).

И наконец, краны стрелового типа оснащаются стрелой, по которой может перемещаться грузовая тележка или на которой может подвешиваться грузозахватный орган.

Несмотря на большую номенклатуру грузоподъемных кранов и разнообразие условий их эксплуатации, они имеют общие основные показатели, по которым может быть произведена оценка их качества, в том числе степень надежности и безопасности.

Надежность и безопасность в зависимости от назначения грузоподъемных кранов и условий их эксплуатации определяются такими показателями, как безотказность, долговечность, срок службы, ремонтпригодность и др. Снижение показателей надежности и безопасности при эксплуатации грузоподъемных кранов приводит к снижению эффективности их работы, преждевременным отказам, повреждениям, несчастным случаям и авариям. Причем число несчастных случаев и аварий, связанных с эксплуатацией грузоподъемных кранов, во времени носит достаточно устойчивый характер, а в ряде случаев и возрастает. Это подтверждает анализ травматизма и аварийности при эксплуатации подъемных сооружений, приведенный далее.

Прежде всего, необходимо отметить, что грузоподъемные краны, так же как и другие подъемные сооружения (подъемники, вышки, краны-манипуляторы, краны-трубоукладчики, лифты), находятся под контролем Федеральной службы по экологическому, технологическому и атомному надзору (Ростехнадзор). При эксплуатации подъемных сооружений должны соблюдаться требования, определенные в различных законах, правилах, по-

ложениях, инструкциях и других нормативных документах, утвержденных Ростехнадзором (в частности, Федеральным законом от 21.07.1997 № 116-ФЗ «О промышленной безопасности опасных производственных объектов»; Правилами устройства и безопасной эксплуатации грузоподъемных кранов (ПБ 10-382-00), с изменениями; Постановлением Госгортехнадзора России от 31.12.99 № 98; приказом Ростехнадзора от 28.10.08 № 849А и др.). Несоблюдение этих требований приводит к авариям, травматизму со смертельным исходом на производстве, в том числе и групповому травматизму со смертельным исходом. Так, в соответствии с «Годовым отчетом о деятельности федеральной службы по экологическому, технологическому и атомному надзору» в 2009 году на предприятиях, где эксплуатируются подъемные сооружения, произошло 28 аварий, материальный ущерб от которых составил 62 млн руб. И хотя это меньше, чем в предшествующие годы (рис. 1.1), и речь идет о 85 519 предприятиях и организациях, контролируемых Ростехнадзором, на которых эксплуатируются 766 103 подъемных сооружения (237 983 грузоподъемных крана, 496 743 лифта, 6 канатных дорог, 3 фуникулера, 7 123 эскалатора, 20 534 подъемника (вышки) и 3 241 грузопассажирский строительный подъемник и подъемник для инвалидов), цифра все равно получается достаточно существенной. Учитывая же общее состояние кранового хозяйства, ситуация представляется весьма серьезной (тем более, что затрагиваются человеческие жизни), а ее ухудшение, к сожалению, более чем вероятно. В 2009 году в 12 авариях из 28 травмировано 16 чел., из них 7 чел. – смертельно. В результате трех аварий (в Республике Татарстан, Тверской и Кемеровской областях) имели место групповые несчастные случаи (для сравнения: в 2008 году было травмировано 24 чел. (погибли 15 чел.), 5 аварий повлекли за собой групповые несчастные случаи).

Наибольшее число аварий в 2009 году произошло при эксплуатации башенных и автокранов (39,3 % общего количества аварий на первых и 32,1 % на вторых). По сравнению с 2008 годом число аварий на автомобильных

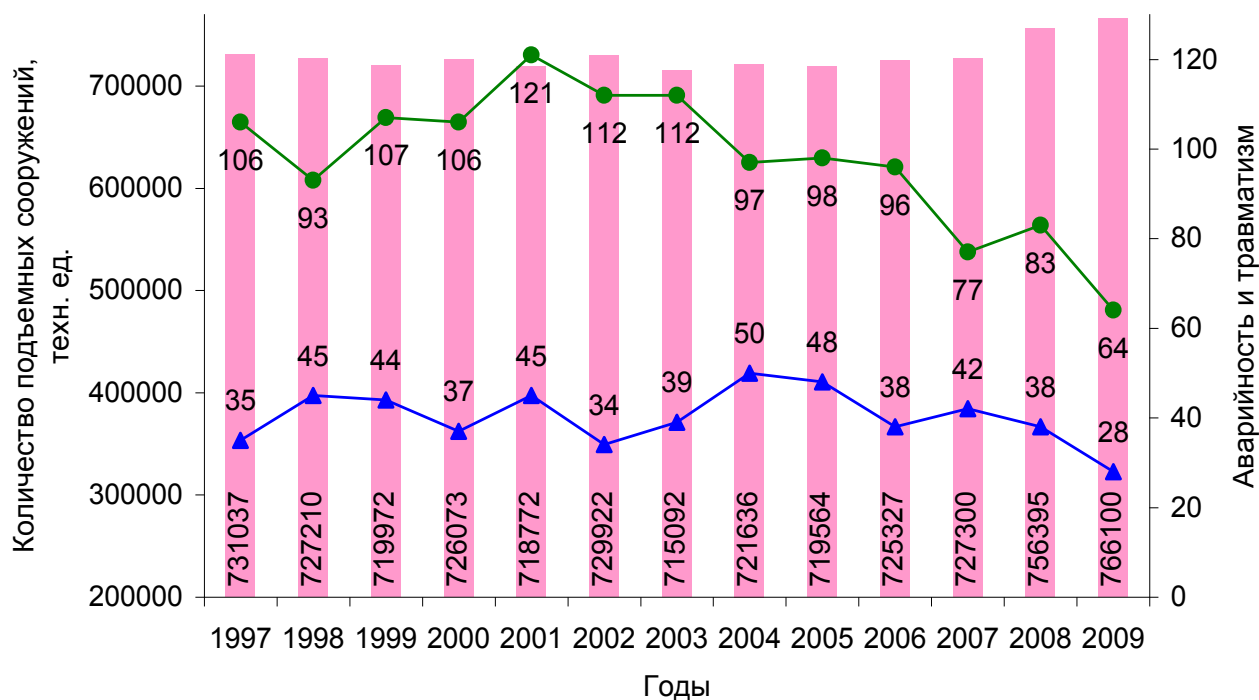


Рис. 1.1. Динамика аварийности и смертельного травматизма при эксплуатации подъемных сооружений:

■ – количество подъемных сооружений, техн. ед.;
 ▲ – число смертельно травмированных; ● – число аварий

кранах уменьшилось на 40 % (с 15 до 9), на гусеничных кранах почти в 2 раза (с 7 до 4), но уровень аварийности на башенных кранах в 2009 году по сравнению с 2008 годом остался на прежнем уровне. Также необходимо отметить, что в 2009 году зафиксировано 3 аварии козловых кранов, в то время как в предшествующем году аварий на этом виде кранов не было (рис. 1.2).

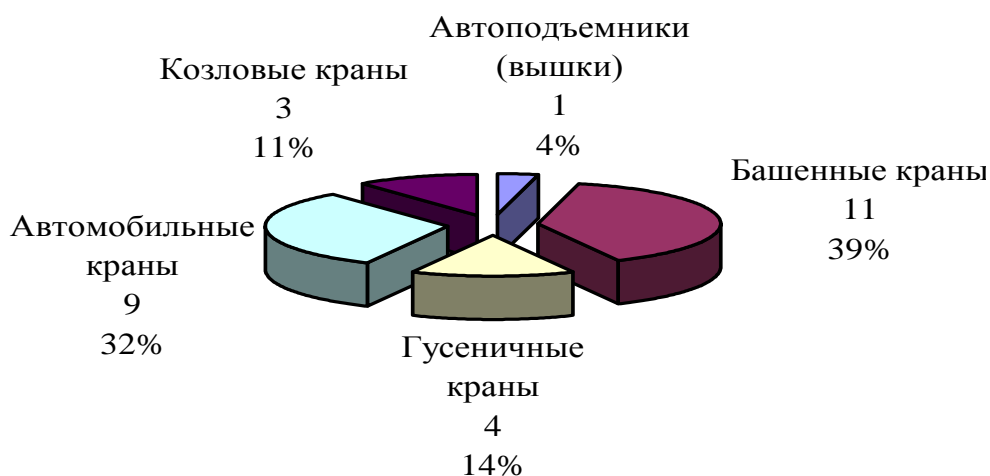


Рис. 1.2. Количество аварий по видам подъемных сооружений в 2009 году

В докладе Ростехнадзора за 2009 год не имеется данных анализа опасных факторов, приведших к возникновению аварий, поэтому приведем их за 2008 год (табл. 1.1).

Таблица 1.1

Опасные факторы и число аварий
на подъемных сооружениях в 2007 и 2008 годах

Опасные факторы	Число аварий		Рост/Снижение (+/-)
	2007	2008	
Некачественное изготовление технических устройств	2	2	–
Неисправность технических устройств	11	13	+ 2
Неисправность приборов безопасности	9	9	–
Неудовлетворительный контроль за соблюдением требований промышленной безопасности	2	1	– 1
Нарушение технологической и трудовой дисциплины, неправильные или несогласованные действия обслуживающего персонала	15	12	– 3
Низкий уровень знаний требований промышленной безопасности	–	–	–
Прочие	3	1	– 2
Всего:	42	38	– 4

Как видно из таблицы, большинство аварий произошло по причине неисправности технических устройств (13) и нарушения технической и трудовой дисциплины (12), причем количество их в 2008 году по сравнению с 2007 годом изменилось незначительно.

В 2009 году при эксплуатации подъемных сооружений было травмировано 119 человек, из них 64 смертельно – на 19 человек меньше, чем за тот же период 2008 года. Тем не менее обращает на себя внимание то, что из 64 несчастных случаев 9 (14 %) произошли при эксплуатации лифтов, 3 (4,7 %) – при применении подъемников (вышек) и 52 (81,3 %) – при эксплуатации грузоподъемных кранов (рис. 1.3).

Анализ несчастных случаев, происшедших при эксплуатации грузоподъемных кранов, позволяет сделать вывод о том, что из всех кранов наиболее высок уровень травматизма при эксплуатации автомобильных (30 % общего числа смертельных случаев на кранах), мостовых (25 %) и башенных (25 %) кранов. При эксплуатации гусеничных кранов в 2009 году погибло 2 человека, что в пять раз меньше, чем в 2008 году. На 41 %

(с 22 до 13 случаев) снизился смертельный травматизм на мостовых кранах, а также на 27 % (с 22 до 16 случаев) – на автомобильных кранах (рис. 1.4).

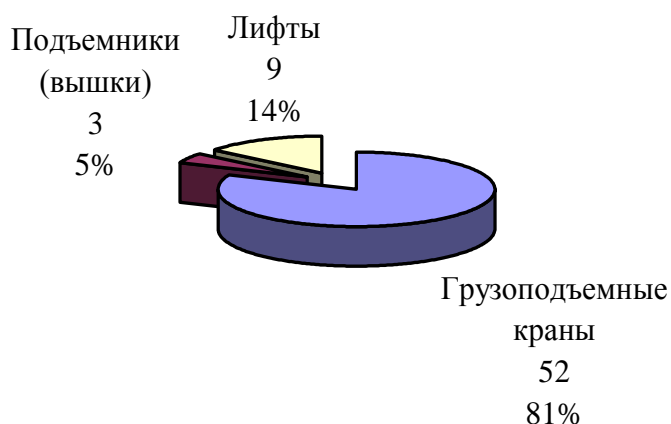


Рис. 1.3. Количество несчастных случаев со смертельным исходом по видам подъемных сооружений в 2009 году

С 15 до 8 уменьшилось количество групповых несчастных случаев. В групповых случаях в 2009 году травмировано 18 человек, из них 9 человек смертельно (в 2008 году – 34 и 11 человек соответственно). Однако в 2009 году резко возросло общее количество несчастных случаев со смертельным исходом, происшедших в результате падения крана из-за его неправильной установки, в результате перегруза и неисправностей. Анализ аварийности и травматизма показал также, что увеличилось количество несчастных случаев, происшедших по вине пострадавшего, находившегося в состоянии алкогольного опьянения, с 1 несчастного случая в 2008 году до 7 случаев в 2009 году (табл. 1.2).

За последние 10 лет количество грузоподъемных кранов в России уменьшилось на 60 тыс. ед. При этом средний коэффициент смертельного травматизма на 1000 кранов по итогам 2009 года составил 0,218 (рис. 1.5). Последнее, казалось бы, создает позитивную перспективу. Вместе с тем, несмотря на наметившуюся положительную тенденцию к уменьшению уровня аварийности и травматизма при эксплуатации кранового оборудования, факт параллельного уменьшения числа кранов при дальнейшем увеличении

объемов и темпов строительства, расширении производственных мощностей и в целом увеличении числа производимых грузоподъемных работ не может не вызывать тревогу. И не в последнюю очередь это связано с крайне медленным обновлением кранового парка: несмотря на ежегодное списание 6–8 тыс. морально и физически устаревших грузоподъемных кранов, темпы ежегодного обновления кранового парка (при норме в 8–10 %) не превышают 1 % в год, что приводит к возрастанию нагрузки на имеющееся крановое

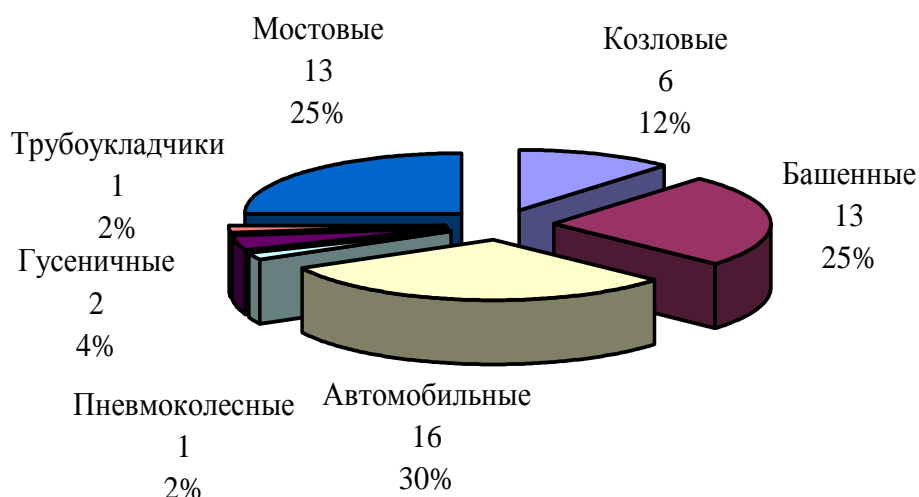


Рис. 1.4. Количество несчастных случаев со смертельным исходом по видам грузоподъемных кранов в 2009 году

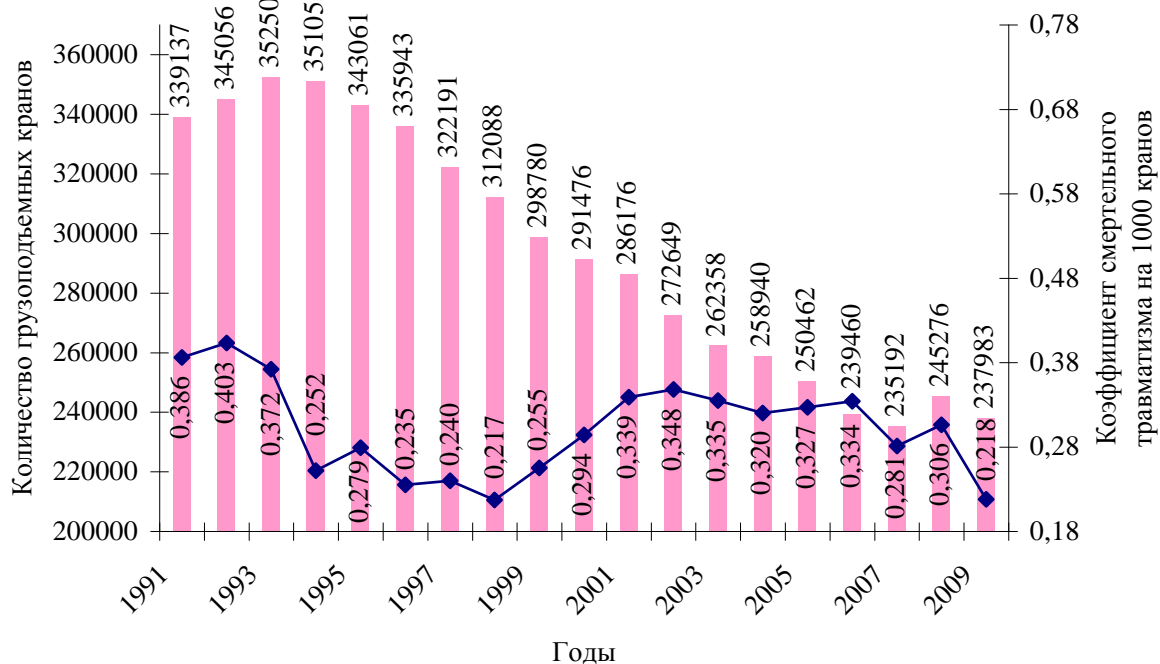


Рис. 1.5. Динамика изменения объема кранового парка России и коэффициента смертельного травматизма

оборудование и увеличению доли грузоподъемных кранов с истекшими сроками службы. Так, по данным Ростехнадзора 80 % грузоподъемных кранов исчерпали нормативные сроки эксплуатации, а около 60 % имеют те или иные опасные дефекты и находятся в предаварийных состояниях.

Таблица 1.2

Причины несчастных случаев со смертельным исходом
на подъемных сооружениях и число погибших в 2008 и 2009 годах

Причина	Число погибших		Рост/Снижение (+/-)
	2008	2009	
Падение груза в результате: применения неисправных или не соответствующих массе и характеру груза грузозахватных приспособле- ний, нарушения схем строповки; нарушения схем складирования грузов	12 2	9 2	- 3 -
Падение крана (стрелы с грузом) в результате: неправильной его установки; перегруза, неисправности приборов безопасности	1 9	6 4	+ 5 - 5
Травмирование: самопроизвольно переместившимся грузом из-за подъема его при наклонном положении грузовых кана- тов (подъем защемленного груза); электрическим током из-за нарушения требований безопасности при работе вблизи ЛЭП; механизмами работающих кранов при выходе людей на крановые пути; грузом, механизмами технических устройств при на- хождении людей в опасной зоне работы кранов	- 7 4 23	- 4 4 12	- - 3 - - 11
Разрушение: кранов или их механизмов из-за содержания техни- ческого устройства в неисправном состоянии; кранов (механизмов) из-за некачественного изготов- ления их на заводе-изготовителе	8 -	1 -	- 7 -
Травмирование механизмами или конструкциями лиф- тов по причине: неисправности лифта или блокировочных устройств; неквалифицированных действий персонала, обслу- живающего лифты; нарушения правил пользования лифтами; проникновения подростков в шахту незаконным образом	2 3 2 -	3 3 3 1	+ 1 - + 1 + 1
Прочие факторы	10	12	+ 2
Всего:	83	64	- 19

Рассмотрим характер этих предаварийных состояний и попытаемся установить их причины более детально.

1.2. Причины аварий и предаварийных состояний грузоподъемных кранов и колебания груза при подъеме и опускании как особо значимые из них

Если грузоподъемный кран изготовлен при строгом соблюдении проекта, технических условий, государственных стандартов и правил безопасности, то он будет выполнять заданные функции, сохраняя свои эксплуатационные качества в установленных пределах в течение установленного срока эксплуатации. Нарушения работоспособности кранов и связанные с ними несчастные случаи и аварии будут вызваны лишь повреждениями и неисправностями, которые возникают при работе кранов с недопустимыми отклонениями от заданных норм их изготовления и эксплуатации.

Работоспособность и техническое состояние крана принято оценивать по наличию опасных дефектов, которые влияют на надежность и безопасность эксплуатации кранов и могут стать причиной внезапных отказов в виде аварий и несчастных случаев. Эксплуатация кранов с такими дефектами недопустима. Если это имеет место, кран должен быть немедленно выведен из эксплуатации для исправления дефектов или списан.

На основе многолетней практики принята классификация, по которой дефекты, выявляемые при техническом диагностировании грузоподъемных машин, по степени их влияния на безопасность эксплуатации делятся на три категории:

- 1-я категория. Дефекты, не влияющие непосредственно на надежность и безопасность эксплуатации грузоподъемной машины. Эксплуатация может быть разрешена без их исправления;
- 2-я категория. Дефекты, снижающие работоспособность и надежность грузоподъемной машины, ее систем, агрегатов и узлов, но не влекущие за собой внезапные отказы в виде аварий. Эксплуатация грузоподъемной машины может быть разрешена на определенный ограниченный период

до исправления дефектов с соблюдением дополнительных мер предосторожности;

- 3-я категория. Дефекты, непосредственно влияющие на надежность и безопасность эксплуатации грузоподъемной машины и влекущие за собой внезапные отказы в виде аварий или несчастные случаи. До устранения этой категории дефектов эксплуатация грузоподъемной машины должна быть прекращена.

Преимуществом приведенной классификации является возможность осуществления регламентации грузоподъемных кранов по степени их опасности при эксплуатации. При этом крайне важно четко квалифицировать дефекты 3-й категории, или опасные дефекты, при наличии которых состояние машины можно оценивать как предаварийное.

В качестве характерных дефектов 3-й категории, которые могут стать причиной аварий или несчастных случаев, можно привести следующие: трещины по сварным швам и основному металлу несущих металлоконструкций; предельные отрицательные прогибы пролетного строения кранов мостового типа; ослабление болтовых соединений скрепления секций башен, стрел и порталов; чрезмерный износ и обрывы проволок грузовых канатов; ненадежное закрепление концов грузовых канатов; предельный износ муфт и тормозных шкивов механизмов подъема; ослабление и отсутствие болтов опорно-поворотных устройств автомобильных, пневмоколесных, гусеничных и башенных кранов; сниженный тормозной момент тормозов механизмов подъема, поворота, изменения вылета, передвижения кранов и тележек; полное или частичное отсутствие либо неработоспособное состояние противоугонных устройств; неработоспособность или неисправность концевых выключателей механизмов; неисправность или отсутствие ограничителей грузоподъемности; отсутствие или неработоспособное состояние анемометров; неудовлетворительное состояние крановых путей; отсутствие или неудовлетворительное состояние тупиковых упоров; неудовлетворительное состояние изоляции силовых кабелей; неработоспособность свето-

вой сигнализации наличия напряжения на главных троллеях мостовых кранов; неработоспособность звуковой сигнализации и др.

Для определения уровня технического состояния грузоподъемных кранов используются заключения экспертизы, проводимой экспертными организациями и специалистами, имеющими соответствующую лицензию. На основании этих заключений выносится решение о дальнейшей эксплуатации крана. Данные экспертизы представляют значительный интерес при проведении анализа (или даже расследования) причин аварий, предаварийных или просто опасных состояний (с возможностью предотвращения аварий), а также при исследовании общей ситуации в крановом хозяйстве региона или организации и изучении имеющихся дефектов в отдельных единицах кранового оборудования с выявлением наиболее опасных из них, определением вероятных причин их возникновения и выработкой мер и технических или организационных решений для предотвращения их образования или уменьшения степени серьезности.

Рассмотрим доступные нам данные за период 2002–2005 гг., приведенные в заключениях экспертных организаций и содержащие информацию о техническом состоянии около 500 единиц грузоподъемных кранов в Енисейском регионе (табл. 1.3, рис. 1.6 и 1.7).

Анализируя эти данные, можно сказать, что обработан достаточный массив кранов для определения картины в целом в регионе и даже в стране, так как считается, что Красноярский край – это уменьшенная копия России и процессы, происходящие в крае, характерны для всей страны.

Из приведенных данных видно, что 346 ед. грузоподъемных кранов из 486, что составляет 71 %, эксплуатируются с опасными дефектами, т. е. эти краны являются потенциальным источником аварийной ситуации. При этом наиболее высок этот показатель у башенных и мостовых кранов (82 и 74 % кранов с опасными дефектами соответственно). На порталных кранах 83 % с опасными дефектами, хотя этот показатель спорен, так как рассмотрены данные по сравнительно небольшому числу этих кранов.

Таблица 1.3

Оценка состояния грузоподъемных кранов по наличию опасных дефектов

Типы кранов	2002 г.		2003 г.		2004 г.		2005 г.		2002–2005 гг.	
	Всего, шт.	С опасными дефектами, шт.	Всего, шт.	С опасными дефектами, шт.	Всего, шт.	С опасными дефектами, шт.	Всего, шт.	С опасными дефектами, шт.	Всего, шт.	С опасными дефектами, шт. (%)
Мостовые	30	7	82	74	74	59	68	47	254	187 (74%)
Козловые	22	5	14	8	11	6	11	7	58	26 (45%)
Башенные	11	7	12	11	9	5	25	24	57	47 (82%)
Стреловые самоходные	16	12	28	18	44	33	23	18	111	81 (73%)
Портальные	2	1	1	1	2	2	1	1	6	5 (83%)
Итого									486	346 (71%)

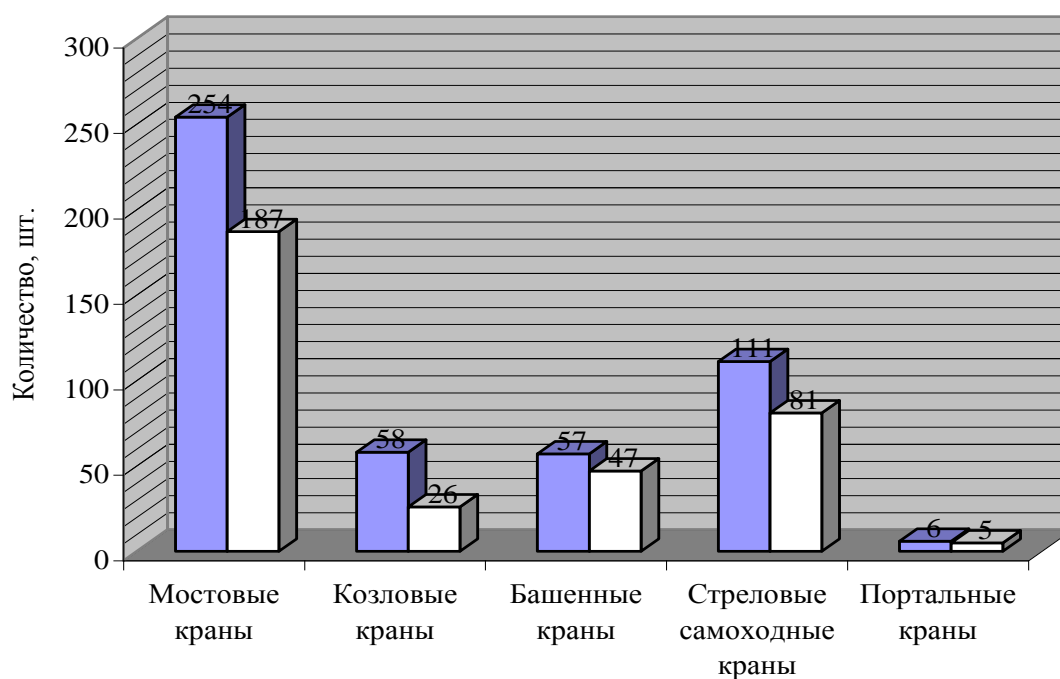


Рис. 1.6. Соотношение числа кранов различных типов с опасными дефектами и общего числа кранов:

■ – общее число; □ – с опасными дефектами

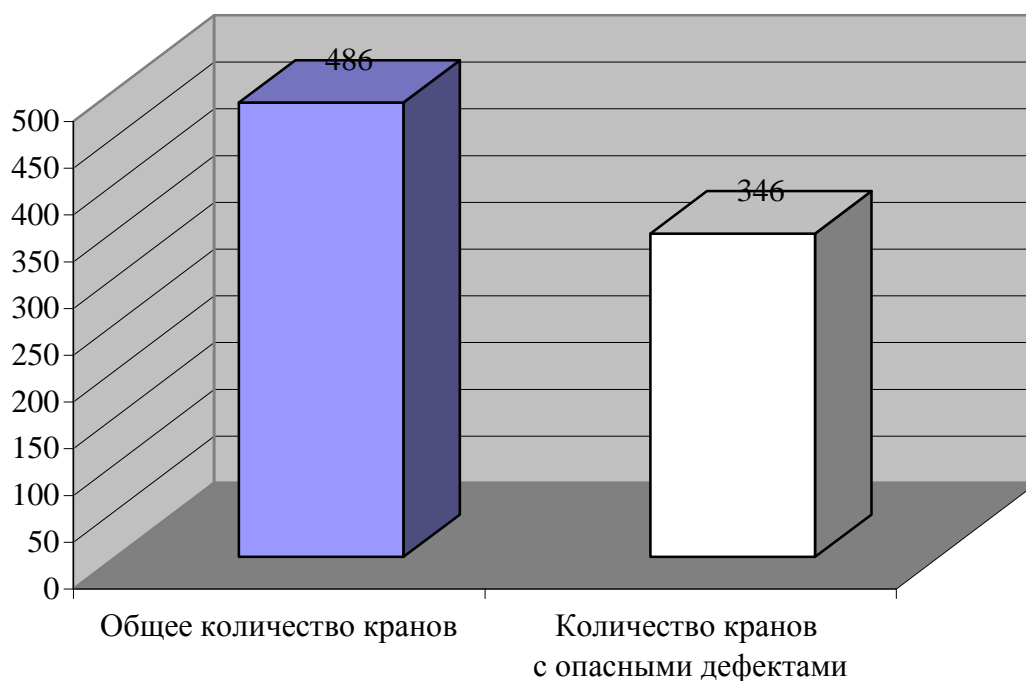


Рис. 1.7. Соотношение общего числа кранов с опасными дефектами и общего числа кранов

Если проанализировать статистические данные о месторасположении выявленных дефектов на кранах мостового типа (мостовых – 254 ед. и козловых – 58 ед.), их описания и фотографии, то нетрудно заметить, как дефекты распределяются по основным составным частям крана, и выделить количество разных опасных дефектов из общего числа (табл. 1.4, 1.5).

Таблица 1.4

Дефекты мостовых кранов, шт.

Год	Металло-конструкции		Механизмы		Электрооборудование		Устройства безопасности		Всего по объекту	
	Всего	Опасные	Всего	Опасные	Всего	Опасные	Всего	Опасные	Всего	Опасные
2002	52	-	152	12	63	3	72	15	339	30
2003	200	21	384	148	213	47	250	84	1047	300
2004	246	112	389	108	107	34	203	65	945	319
2005	136	42	386	164	95	31	256	116	873	353
Всего:	634	175	1311	432	478	115	781	280	3204	1002

У мостовых кранов можно отметить превалирование дефектов, обнаруженных в механизмах крана. Характерные дефекты механизмов: неправильная регулировка тормозов механизмов передвижения моста крана и тележки; предельные износы тормозных накладок механизмов передвижения моста и грузовой тележки; предельно допустимый износ реборд ходовых колес крана и тележки; ослабление и отсутствие болтов на муфтах механизмов передвижения и подъема; отсутствие защитных кожухов на муфтах; отсутствие пружинных шайб на прижимных планках каната на барабане. Некоторое снижение числа дефектов в 2005 году связано с меньшим числом обследованных кранов, при этом количество опасных дефектов максимально, что может свидетельствовать о старении кранового оборудования и переходе дефектов, изначально классифицированных как «безопасные», в категорию «опасных».

Характерными дефектами металлоконструкций, выявленными на мостовых кранах в ходе обследований, являются: деформация поперечной балки рамы грузовой тележки от переподъема крюковой подвески (рис. 1.8); трещины в местах крепления бус на концевых балках; трещины в местах стыка главных и концевых балок; неудовлетворительное состояние окраски и начальная коррозия металлоконструкций в местах отслоения и шелушения краски (рис. 1.9); трещины рамы тележки в районе крепления редуктора механизма передвижения тележки; трещины в поперечных балках рамы грузовой тележки в местах приварки косынок; трещины в околобусовых зонах на грузовой тележке (рис. 1.10).

Характерные неисправности электрооборудования: отсутствие дугогасительных камер; отсутствие знаков опасного напряжения на шкафах электрооборудования и на защитной панели; не заделаны вводы проводов электрооборудования; вместо предохранителей в защитной панели установлены перемычки.



Рис. 1.8. Деформация поперечной балки рамы грузовой тележки от переподъема крюковой подвески



Рис. 1.9. Неудовлетворительное состояние окраски и начальная коррозия металлоконструкций в местах отслоения и шелушения краски



Рис. 1.10. Трещины в околобуксовых зонах на грузовой тележке

Характерные неисправности приборов и устройств безопасности: неправильная регулировка выключающей линейки, из-за чего не срабатывают концевые выключатели механизмов передвижения моста крана; выведены из действия блокировки на двери кабины крановщика и люке на мосту крана.

Таблица 1.5

Дефекты козловых кранов, шт.

Год	Металло-конструкции		Механизмы		Электрооборудование		Устройства безопасности		Всего по объекту	
	Всего	Опасные	Всего	Опасные	Всего	Опасные	Всего	Опасные	Всего	Опасные
2002	60	18	81	25	26	12	34	18	201	73
2003	72	4	53	11	22	7	38	13	185	35
2004	33	1	44	6	10	2	32	11	119	20
2005	24	4	48	12	14	5	27	13	113	34
Всего:	189	27	226	54	72	26	131	55	618	162

В отличие от ситуации по мостовым кранам, количество дефектов в механизмах козловых кранов лишь незначительно превышает их количество в металлоконструкциях и устройствах безопасности (хотя в металлоконструкциях дефектов больше, чем в указанных устройствах). Однако обращает на себя внимание опять более существенная цифра опасных дефектов по 2005 году (меньшее общее число по сравнению с 2002, по-видимому, вызвано меньшим числом обследованных кранов).

Характерными дефектами металлоконструкций на обследованных козловых кранах являются: неудовлетворительное состояние окраски и начальная коррозия металлоконструкций в местах отслоения и шелушения краски и в местах скопления влаги; трещины и деформации в стяжках стоек опор (рис. 1.11); трещины в месте соединения стойки опоры с аркой; трещины по сварным швам в местах крепления стяжек с рамами ходовых тележек; трещины в раме грузовой каретки (рис. 1.12); деформации раскосов пролетного строения (рис. 1.13).



Рис. 1.11. Трещины и деформации в стяжках стоек опор (ККС-10)

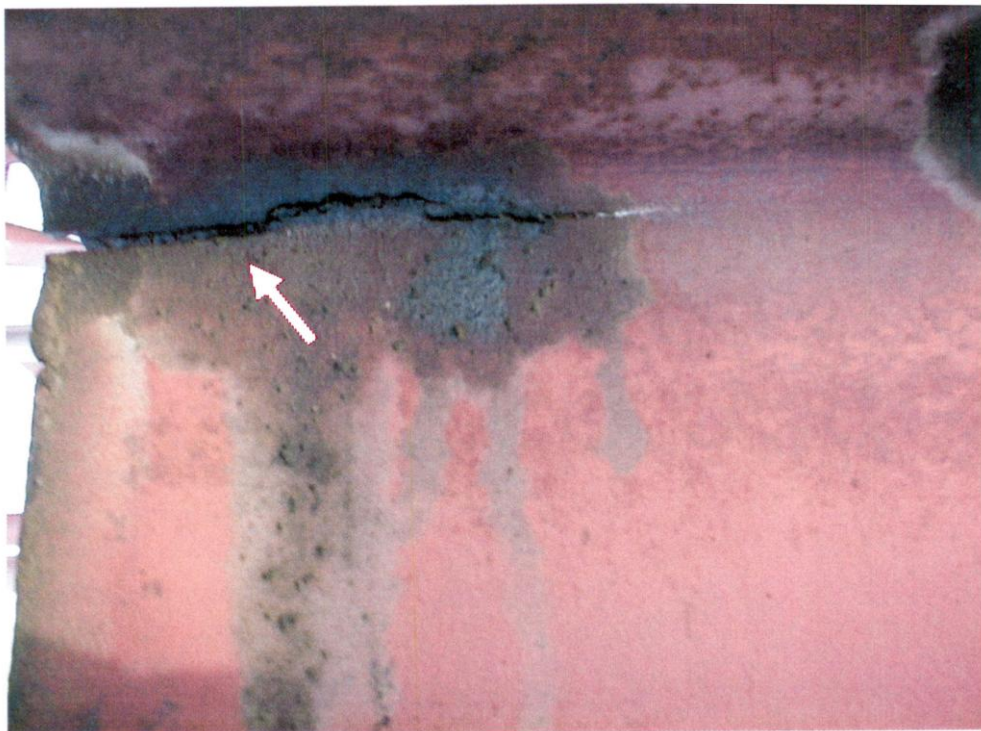


Рис. 1.12. Трещины в раме грузовой каретки (ККС-10)



Рис. 1.13. Деформации раскосов пролетного строения

Характерные дефекты механизмов: неправильная регулировка тормозов механизмов передвижения крана и грузовой тележки; подтекание масла

из редукторов, подтекание рабочей жидкости из гидротолкателей; предельно допустимый износ реборд ходовых колес крана.

Характерные неисправности электрооборудования: нарушение изоляции проводки, неисправности в системе освещения; нет заземления электронагревательных печей в кабине крановщика; не заделаны вводы проводов электрооборудования.

Характерные неисправности приборов и устройств безопасности: выведены из действия автоматические противоугонные захваты; неисправны концевые выключатели механизмов передвижения крана и передвижения грузовой тележки; ходовые колеса крана и тележки не снабжены щитками, предотвращающими возможность попадания под колеса посторонних предметов; нет буферов на ходовых тележках крана.

Все изложенное позволяет составить общую картину состояния кранового оборудования и дает представление о наиболее характерных дефектах и неисправностях, позволяя выявить причины их возникновения. Анализируя эту картину, а также руководствуясь дополнительными материалами, можно свести имеющиеся данные в общую таблицу, в которой основным существующим дефектам и неисправностям кранового оборудования правомерно поставить в соответствие возможные причины их возникновения (табл. 1.6).

Таблица 1.6

Возможные причины основных дефектов
и неисправностей грузоподъемных кранов

Дефект или неисправность	Возможные причины
Металлоконструкции	
Трещины в местах крепления угловых букс	Динамические нагрузки (наиболее напряженная часть балки, воспринимающая вертикальные усилия)
Трещины в концевых балках	Динамические нагрузки (при конструктивных недостатках крепления узла ходового колеса); неудовлетворительное качество сварных работ

Дефект или неисправность	Возможные причины
Трещины верхнего пояса главных балок	Динамические нагрузки (многократное действие в поясе под стыком подтележного рельса повышенных (при зазорах между поясом и диафрагмами и разрушении сварных швов) местных напряжений от катковой (двигательной) нагрузки)
Трещины в местах резкого изменения поперечного сечения главной балки	Динамические нагрузки (высокая концентрация напряжений в местах перегиба пояса сварного шва); образование при сборке щелей и заварка их наплавкой дополнительного металла
Трещины по угловому шву	Динамические нагрузки (нижний пояс концевой балки является накладкой для нижнего пояса главной балки)
Деформация пролетных балок мостовых и козловых кранов в вертикальной плоскости (отрицательный прогиб)	Динамические нагрузки (при циклическом характере нагружения балок); состояние крановых путей
Разрушение продольных связей в местах их крепления к торцевому листу концевой балки	Динамические нагрузки (возникновение при колебаниях балки изгибных напряжений в местах крепления продольных элементов к торцевому листу при недостаточной жесткости из-за значительной гибкости продольных связей)
Трещины в элементах крепления механизмов моста крана и грузовой тележки	Динамические нагрузки (в момент пуска двигателя в узлах крепления редуктора возникают значительные усилия, передающиеся на ребро и стенку в виде изгибающего момента)
Трещины и обрывы кронштейнов (площадок, на которых установлены приводы) в местах их приварки к вертикальной стенке пролетной балки	Недостаточная надежность конструкций при значительных нагрузках
Повреждения металлоконструкций мостовых кранов в виде язвенной коррозии и трещин в местах наличия полостей («карманов»), щелей	Неплотное прилегание элементов из-за большого шага заклепок или сварки прерывистым швом
Коррозионные повреждения внутренних поверхностей коробчатых балок при длительном хранении кранов или эксплуатации на открытом воздухе	Скопление влаги в местах крепления площадок и на верхних поясах
Коррозийные повреждения в местах щелей, имеющих в составных элементах, узлах соединения пролетного строения с опорами	Скопление влаги в местах приварки фасонных накладок; динамические нагрузки (при их возрастающем влиянии и превышении)
Неудовлетворительное состояние окраски и начальная коррозия металлоконструкций в местах отслоения и шелушения краски	Несоблюдение норм эксплуатации

Продолжение табл. 1.6

Дефект или неисправность	Возможные причины
Механизмы	
Износ зубчатых передач	Динамические нагрузки (широкий диапазон нагрузок (от минимальных до максимальных), зачастую значительно превышающих номинальные и не зависящих от массы поднимаемого груза); недостаток смазки
Износ дорожек качения	Недостаток смазки; неправильная установка подшипников
Выход из строя подшипников, соединительных муфт и валов	Неправильная установка валов; динамические нагрузки
Неравномерный износ тормозных шкивов, перекос тормоза	Неправильная регулировка тормоза
Ослабление пружины тормоза	Несоблюдение норм эксплуатации; динамические нагрузки
Перекос крана относительно рельсов кранового пути, износ реборд, выход из строя ходовых колес	Неправильная установка колес относительно моста
Сход колес тележки и моста с рельсов	Попадание посторонних предметов
Заклинивание тележки при движении ее на мосту крана	Сужение (уширение) колеи подтележных рельсов
Подтекание масла из редукторов, подтекание рабочей жидкости из гидротолкателей	Несоблюдение норм эксплуатации (недостаточная затяжка болтов); выход из строя уплотнения
Пробуксовка ходовых колес	Установка двигателя завышенной мощности
Перетирание и повреждение канатов	Установки канатов на барабане больше нормы по длине; смещения к центру барабана креплений их концов; защемление между блоками; задевание их при подъеме груза и передвижении тележки; динамические нагрузки
Повреждение обоймы и кожуха	Использование тонкого стального листа; защемление каната
Сползание блоков	Ослабление пружинного кольца
Ослабление болтовых соединений скрепления секций башен, стрел и порталов	Динамические нагрузки
Ненадежное закрепление концов грузовых канатов	Несоблюдение норм эксплуатации
Электрооборудование	
Неудовлетворительное состояние изоляции силовых кабелей	Несоблюдение норм эксплуатации
Отсутствие дугогасительных камер	Несоблюдение норм эксплуатации
Отсутствие знаков опасного напряжения на шкафах электрооборудования и на защитной панели	Несоблюдение норм эксплуатации

Дефект или неисправность	Возможные причины
Устройства безопасности	
Полное или частичное отсутствие либо неработоспособное состояние противоугонных устройств	Несоблюдение норм эксплуатации
Неработоспособность или неисправность концевых выключателей механизмов	Несоблюдение норм эксплуатации
Неисправность или отсутствие ограничителей грузоподъемности	Несоблюдение норм эксплуатации
Отсутствие или неработоспособное состояние анемометров	Несоблюдение норм эксплуатации
Отсутствие или неудовлетворительное состояние тупиковых упоров	Несоблюдение норм эксплуатации
Неработоспособность световой и звуковой сигнализаций	Несоблюдение норм эксплуатации

Анализируя табл. 1.6, можно отметить превалирование (и разнообразие) дефектов и неисправностей, локализованных, прежде всего, в области механизмов и металлоконструкций. С другой стороны, обобщая, можно выделить четыре основных вида причин появления дефектов в крановом оборудовании: динамические нагрузки, несоблюдение норм эксплуатации, неправильное конструктивное исполнение, неправильная установка и сварка. При этом первые два непосредственно или опосредованно влияют на возникновение большинства неисправностей не только механизмов и металлоконструкций.

Если методы борьбы с несоблюдением норм эксплуатации представляются очевидными и нуждающимися лишь в последовательном воплощении, то имеющиеся сегодня технические решения в направлении ослабления влияния динамических нагрузок пока еще далеки от идеала.

Динамические (инерционные) нагрузки возникают в периоды неустановившегося движения при разгоне и торможении крановых приводов, а также вследствие толчков и ударов. Их следствием, а зачастую и причиной являются горизонтальные раскачивания и вертикальные колебания груза при его подъеме и опускании. Наиболее существенное влияние колебания оказывают на металлоконструкции кранов. Анализ отказов узлов металло-

конструкций мостовых кранов показывает, что в первую очередь повреждаются наиболее нагруженные элементы. Так, 62 % повреждений возникают в местах крепления угловых боксов (рис. 1.14). Узел крепления угловых боксов является наиболее напряженной частью балки, которая воспринимает вертикальные усилия, возникающие при движении крана. Трещина 3, например, появляется в криволинейной части перехода от меньшего сечения к большему. Очень часто трещины от сварного шва распространяются на основной металл. Так, трещины 5 и 6 формируются в углах вырезов, поражая затем основной металл стенки. Образование трещин, подобных трещине 2, и последующее разрушение металла начинается в местах приварки ребра к поясу балки, а затем распространяется на сварной шов соединения стенки с поясом и на основной металл. Зачастую очагами начального разрушения балки являются ребра жесткости 4. Трещины, подобные трещине 1, образуются в вертикальной стенке на расстоянии 100–150 мм от вертикального пояса в местах установки диафрагм.

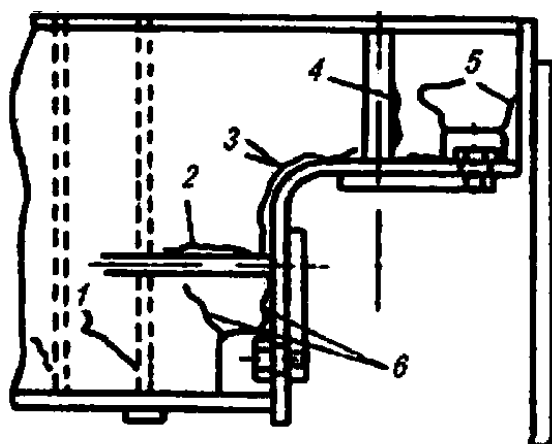


Рис. 1.14. Трещины в местах крепления угловых боксов

При эксплуатации мостовых и козловых кранов постоянно циклически деформируются в вертикальной плоскости пролетные балки. В начальной стадии деформация влечет за собой уменьшение строительного подъема, вплоть до его исчезновения. В дальнейшем балка приобретает постоянно увеличивающийся так называемый отрицательный прогиб.

На кранах, имеющих перильные фермы, после непродолжительной эксплуатации происходит разрушение продольных связей в местах их крепления к торцевому листу концевой балки. Разрушение начинается с появления на расстоянии 50–100 мм от сварного шва трещины, которая приводит к обрыву связей. Узел крепления фермы с концевыми балками не обладает необходимой жесткостью из-за значительной гибкости продольных связей, поэтому при колебаниях балки в местах крепления продольных элементов к торцевому листу возникают изгибные напряжения. Сварные соединения в этих местах увеличивают концентрацию напряжений, а это ведет к быстрому разрушению крепления.

Повреждения металлоконструкций мостовых кранов в виде язвенной коррозии и трещин наблюдаются в местах наличия полостей («карманов»), щелей, возникающих при сварке прерывистыми швами, или в результате неплотного прилегания элементов из-за большого шага заклепок. В концевых балках мостовых кранов влага скапливается в местах крепления площадок и на верхних поясах. При длительном хранении кранов или эксплуатации на открытом воздухе происходят коррозионные повреждения внутренних поверхностей коробчатых балок. В таких условиях процесс коррозионного разрушения и сопровождается трещинами. Он имеет три характерные стадии: медленное развитие трещины, которое в основном определяется непосредственно коррозионным фактором; скачкообразное, относительно быстрое развитие трещины при возрастающем влиянии динамической нагрузки; лавинообразное разрушение (разрыв) при превышении нагрузки.

Несмотря на разнообразие описанных выше трещин, все они, как правило, – это следствие усталостного разрушения элементов кранов, которое обусловлено, прежде всего, упоминавшимися ранее переменными нагрузками, вызываемыми горизонтальными раскачиваниями и вертикальными колебаниями перемещаемых грузов. Раскачивания грузов и колебания особенно заметны при торможении механизмов кранов (подъема груза, передвижения тележки, поворота стрелы и т. п.). Однако если раскачивания и

способы их устранения изучены в значительной степени, то вертикальные колебания при подъемах – опусканиях груза на данный момент представляются более проблемными, поскольку изучены меньше.

В связи с изложенным в течение многих лет была и остается актуальной задача совершенствования существующих и создания новых технических средств гашения колебаний грузов при эксплуатации кранов. Некоторые из путей решения такой задачи на основе исследования вертикальных колебаний и анализа существующих технических средств, предназначенных для их устранения, и рассматриваются в настоящей работе.

2. СОВРЕМЕННЫЕ АВТОМАТИЧЕСКИЕ СРЕДСТВА УСТРАНЕНИЯ ВЕРТИКАЛЬНЫХ КОЛЕБАНИЙ ГРУЗОВ, ПЕРЕМЕЩАЕМЫХ КРАНОМ

2.1. Устройства активного виброгашения с измерением частоты вынужденных колебаний

Одним из наиболее эффективных автоматических средств устранения вертикальных колебаний грузов, перемещаемых кранами, в настоящее время являются активные (адаптивные) виброгасители, включающие в себя специальные блоки, предназначенные для измерения частоты вынужденных колебаний груза, закрепленного на крюковой подвеске крана. Применить такие виброгасители при эксплуатации грузоподъемных кранов было предложено в работе А. М. Кабакова, А. И. Пабата и А. Н. Орлова, опубликованной в [2]. Виброгасители, описанные в указанной работе, содержат датчик частоты колебаний, устанавливаемый на крюковой подвеске, управляющее устройство и собственно виброгаситель, устанавливаемые там же. Последний имеет в своем составе исполнительные элементы, образующие регулятор частоты собственных колебаний, который, будучи соединен с управляющим устройством, подстраивает эту частоту в резонанс с частотой вынужденных колебаний, измеряемой датчиком. Подстройка в резонанс обеспечивает практически полное поглощение энергии колебаний подвески с грузом, что и способствует их эффективному гашению.

Способов адаптивного виброгашения сегодня существует несколько, что обусловлено зависимостью [3]

$$f = \frac{1}{4\pi m} \sqrt{4mc - k^2}, \quad (1)$$

где f – частота собственных колебаний виброгасителя, Гц; m – его рабочая масса, кг; c – жесткость, Н/м; k – коэффициент вязкого трения, кг/с; в гид-

равлических виброгасителях, содержащих дроссели, равный, в частности,

$$k = \frac{\mu F}{10^3 h}, \quad (2)$$

где μ – вязкость жидкости, используемой в виброгасителе, Па·с; F – площадь поверхности щели его дросселя, мм²; h – ширина щели, мм.

Регулировать f , подстраивая ее под частоту вынужденных колебаний, в соответствии с выражениями (1) и (2) можно путем изменения всех входящих в них величин. Это порождает различные конструктивные схемы адаптивных виброгасителей, наиболее характерные из которых приведены ниже.

На рис. 2.1 показана практически первая из существующих схем рассматриваемых виброгасителей [4], согласно которой на штоке 1 с помощью мембраны 2 закреплен груз 3, подтягиваемый вверх жестко закрепленным на штоке электромагнитом 4. Вместе с электромагнитом на штоке установлен датчик частоты 5, соединенный с электромагнитом через устройство управления 6 и аналоговый сумматор. Виброгаситель ввинчивается в вибрирующий объект 7 так, чтобы его ось совпадала с направлением вибрации, а затем путем подачи некоторого напряжения смещения на клемму 8 настраивается. Сила притяжения электромагнита частично компенсирует силу тяжести груза (уменьшает его статическую осадку), что в определенной степени эквивалентно уменьшению его массы. При изменении частоты вынужденных колебаний датчик частоты изменяет величину компенсации и частота собственных колебаний виброгасителя подстраивается в резонанс.

На рис. 2.2 показан виброгаситель [5], который также работает с изменением массы, но иным способом. В его корпусе 1 на упругой мембране 2 закреплен груз 3, выполненный в виде пружинного гидроаккумулятора. Датчик частоты 4, установленный на корпусе 1, через устройство управления 5 подключен к электроуправляемому дросселю 6. Через постоянный (настроечный) дроссель 7 в полость гидроаккумулятора и на дроссель 6 под давлением P подается жидкость. Она приподнимает поршень 8, сжимая пружину 9 гидроаккумулятора, и заполняет определенный объем последне-

го, создавая его определенную массу. По сигналу от датчика 4 дроссель 6 регулирует слив жидкости в емкость 10, тем самым изменяя массу жидкости в гидроаккумуляторе 3. Получается, что это изменение задается частотой вынужденных колебаний объекта 11, вибрации которого требуется погасить. Таким образом, частота собственных колебаний виброгасителя все время согласуется с частотой вынужденных колебаний объекта.

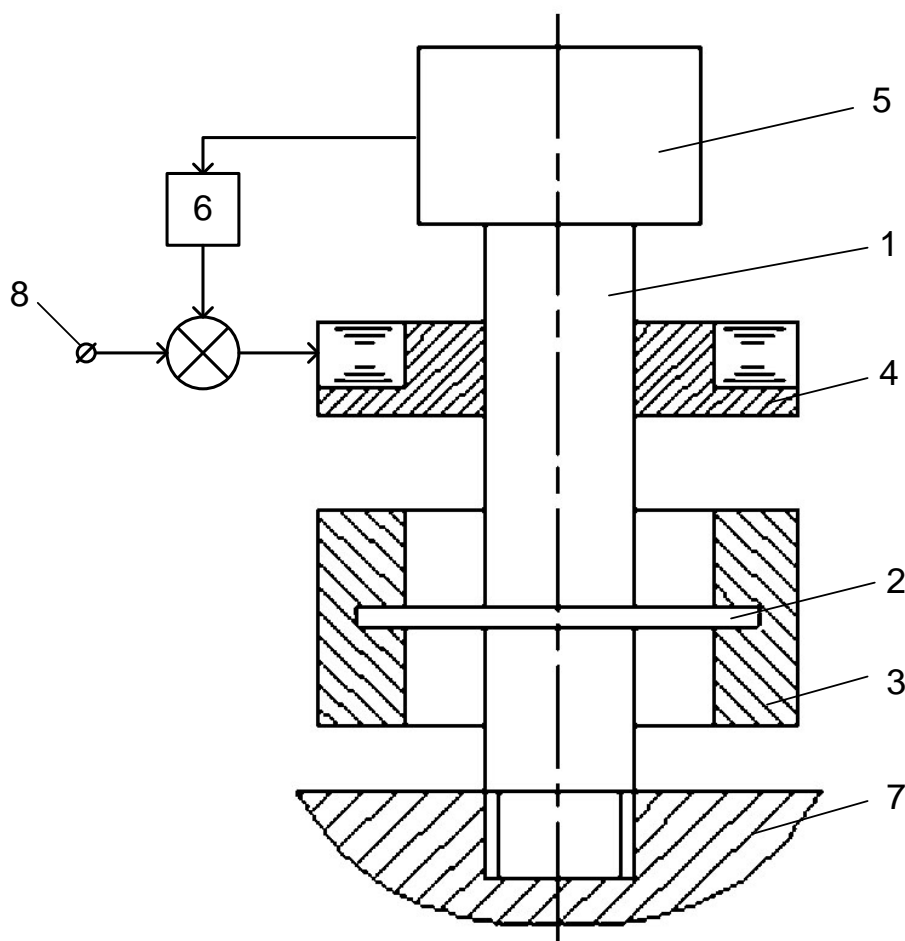


Рис. 2.1. Виброгаситель по а. с. СССР № 518589

Конструктивные схемы виброгасителей, работающих с изменением жесткости, приведены на рис. 2.3 и 2.4. Первая из них [6] содержит корпус 1, мембрану 2 с грузом 3 аналогично предыдущему варианту, но ее особенностью является применение пневмоцилиндров 4. Датчик частоты 5, установленный на корпусе 1, через устройство управления 6 в этом виброгасителе соединен с регулятором давления воздуха 7, питающим пневмоцилиндры. Воздух подается в бесштоковые полости цилиндров и изменяет

фактическое положение точек защемления мембраны относительно кронштейнов 8, на которых мембрана крепится в корпусе 1. Это происходит в результате перемещения губок 9 штоков поршней 10 относительно кронштейнов 8 в окнах 11 штоков и смещения штоков в отверстиях 12. Обратное перемещение поршней осуществляется под действием пружин 13 при стравливании воздуха из цилиндров.

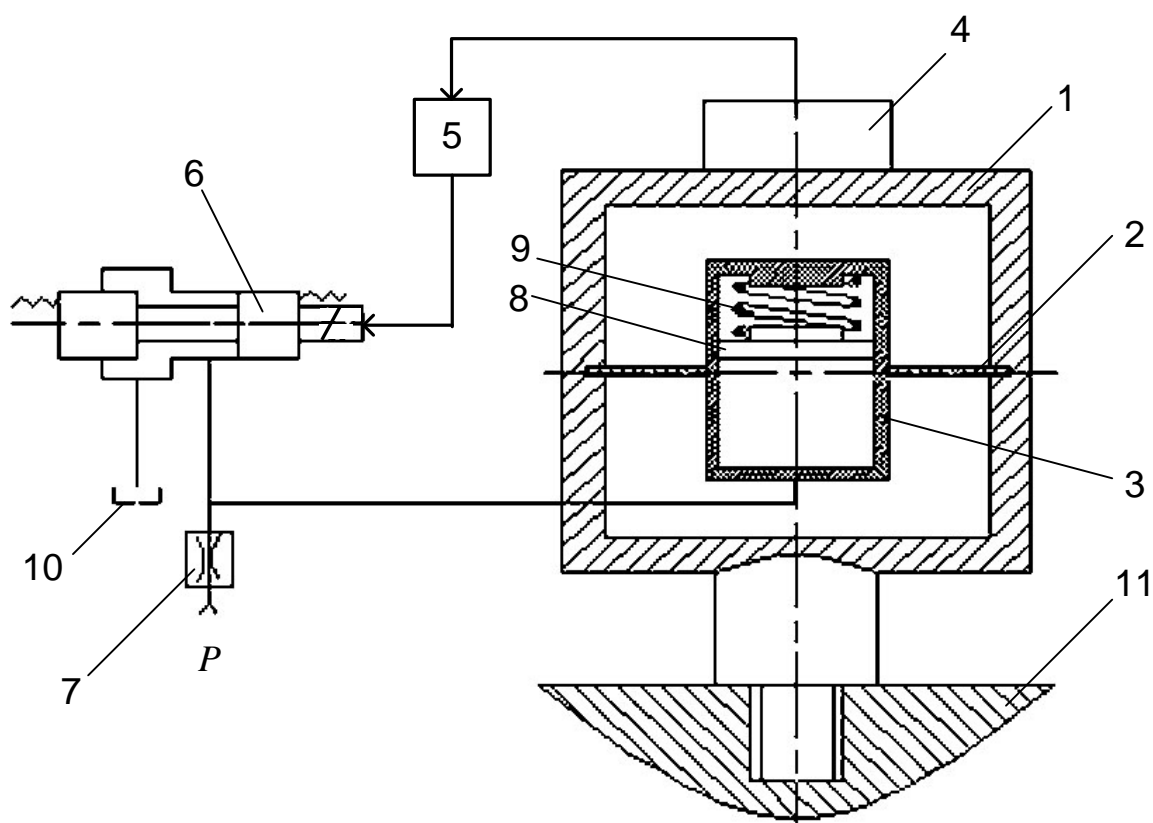


Рис. 2.2. Виброгаситель по патенту РФ № 87482

Вторая схема управления частотой собственных колебаний виброгасителя путем изменения жесткости [7], иллюстрируемая рис. 2.4, основывается на изменении упругости жидкости при переменном давлении. Как известно [8], во всякой жидкости содержатся нерастворенные газы. Их количество определяет ее модуль упругости (см. пример на рис. 2.5). В связи с этим жидкость в некоторой емкости может рассматриваться как особого рода пружина с регулируемой жесткостью. Корпус 1 виброгасителя, построенного на таких пружинах, так же как и в ранее рассмотренных случаях,

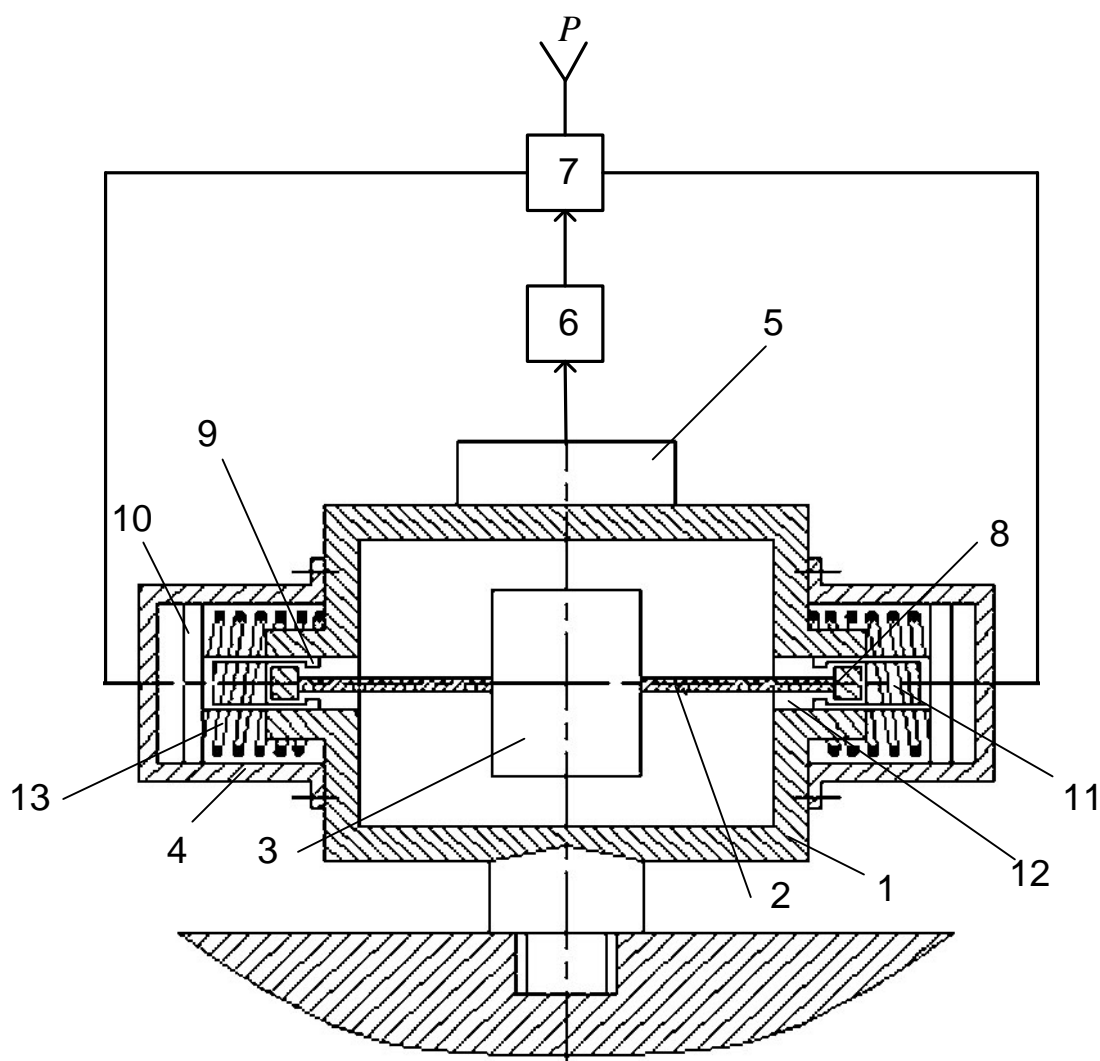


Рис. 2.3. Виброгаситель по патенту РФ № 87484

ввинчивается в вибрирующий объект 2. Внутри корпуса размещаются датчик частоты 3 и груз 4. Датчик через устройство управления 5 соединяется с регулятором давления жидкости 6. Груз прикрепляется к корпусу с помощью сильфонов 7. Полости сильфонов заполнены жидкостью, поступающей извне через регулятор 6. Меняется частота вынужденных колебаний – соответственно меняется давление в сильфонах, их жесткость и частота собственных колебаний.

Описанный виброгаситель подобен показанному на рис. 2.6. Однако принцип действия последнего другой [9]. В его корпусе 1 по скользящей посадке размещен груз 2, поджатый с двух сторон пружинами 3.

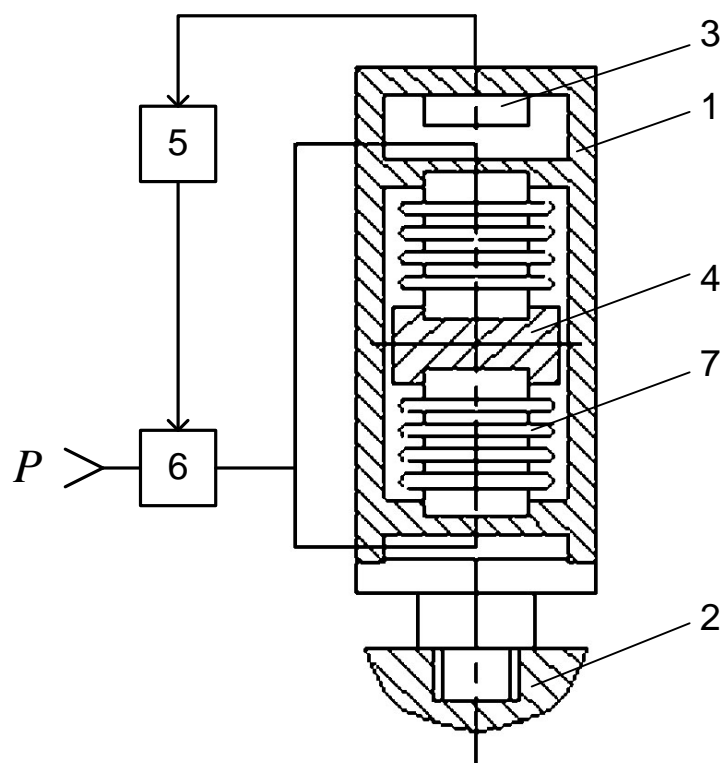


Рис. 2.4. Виброгаситель по патенту РФ № 87483

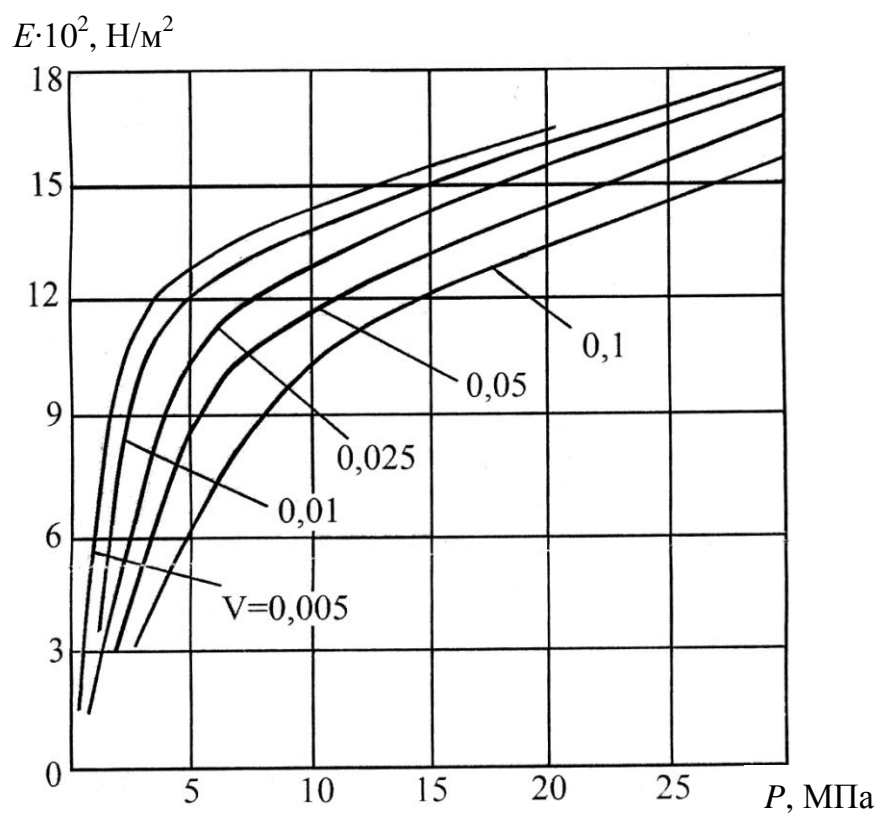


Рис. 2.5. Зависимость модуля объемной упругости E от давления P для минерального масла АМГ-10 при различных V – отношениях объема нерастворенного газа к объему жидкости

Груз делит полость корпуса на две части, которые заполняются вязкой жидкостью, например машинным маслом. На корпусе 1 закреплен датчик частоты 4, соединенный с электроуправляемым дросселем 5 через устройство управления 6. Обе части полости корпуса связаны между собой через этот же дроссель, и от ширины его щели зависит, как будет перетекать масло из одной части полости в другую при колебаниях груза 2. Однако ширина щели регулируется частотой вынужденных колебаний, поэтому, установив виброгаситель на колеблющемся объекте 7, можно обеспечить управление частотой собственных колебаний груза в согласии с частотой колебаний вынужденных.

Управление шириной дросселирующей щели – это, по сути, управление коэффициентом вязкого трения в формуле (1). Однако, как видно из (2), этот коэффициент можно изменять и за счет вязкости масла. Вязкость же масла существенно зависит от его температуры [8], поэтому, окружив масляный виброгаситель водяной рубашкой и меняя температуру в ней, можно регулировать и коэффициент вязкого трения (см. пример на рис. 2.7), и частоту собственных колебаний виброгасителя.

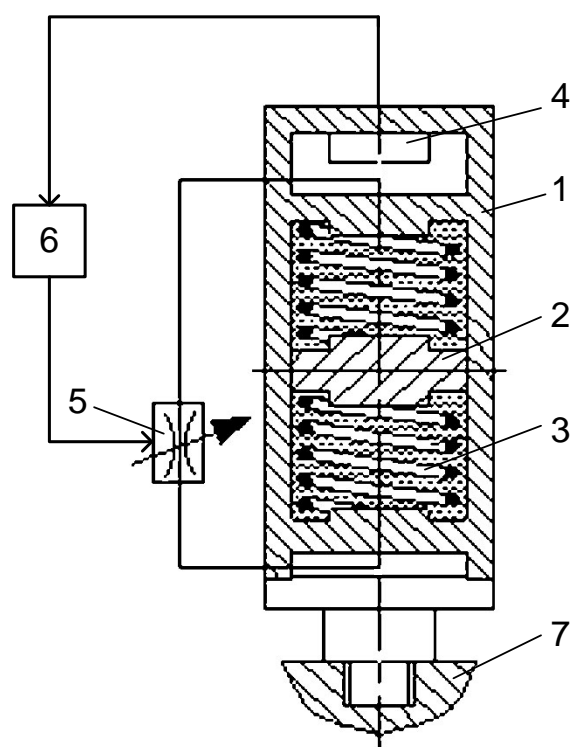


Рис. 2.6. Виброгаситель по патенту РФ № 87481

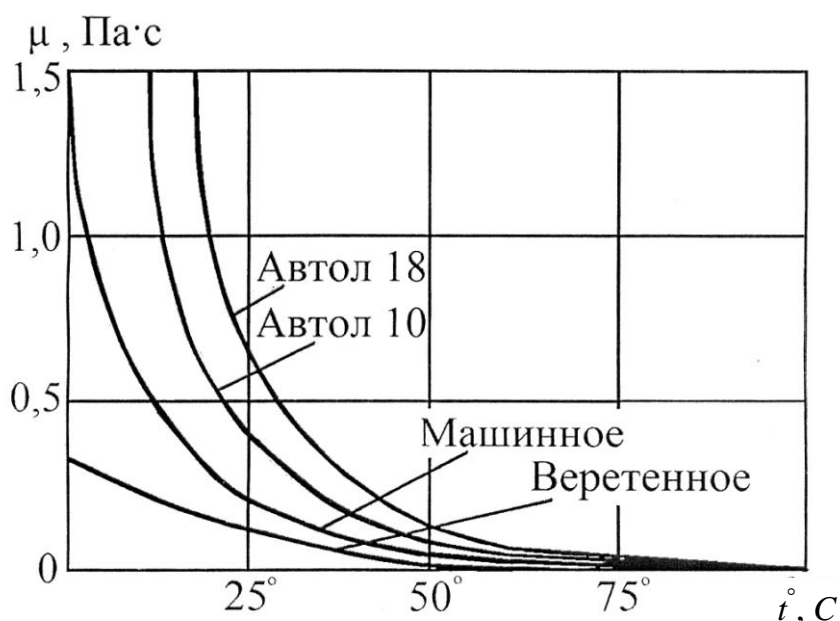


Рис. 2.7. Зависимость вязкости μ различных масел от температуры t

Подобное решение предложено в [10] и иллюстрируется рис. 2.8. Здесь 1 – корпус виброгасителя, так же как и всегда, ввинчиваемый в вибрирующий объект 2, 3 – датчик частоты. Помимо этого, виброгаситель снабжен терморегулятором жидкости 4 и прокачивающим через него воду насосом 5. В корпусе 1 с зазором размещен груз 6, поджатый с двух сторон пружинами 7. В грузе выполнены дросселирующие отверстия, корпус заполнен маслом и окружен водяной рубашкой 8, соединенной кольцевой гидролинией с терморегулятором 4 и насосом 5. Датчик 3 через устройство управления 9 и сумматор напряжений 10 подключен к управляющему входу терморегулятора, поэтому при изменении частоты вынужденных колебаний ранее настроенная сигналом A температура воды в рубашке будет меняться, а значит, изменится и вязкость масла в полости корпуса 1, и частота собственных колебаний виброгасителя.

Описанные конструктивные схемы и принципы построения активных (адаптивных) виброгасителей в настоящее время могут быть практически реализованы по-разному. Оригинальный вариант гидравлических виброгасителей с регулируемой вязкостью используемой в нем жидкости предложен, например, в [2]. Вместо обычного масла в нем применена магниторео-

логическая суспензия, изменяющая вязкость в магнитном поле, а вместо водяной рубашки использован электромагнит.

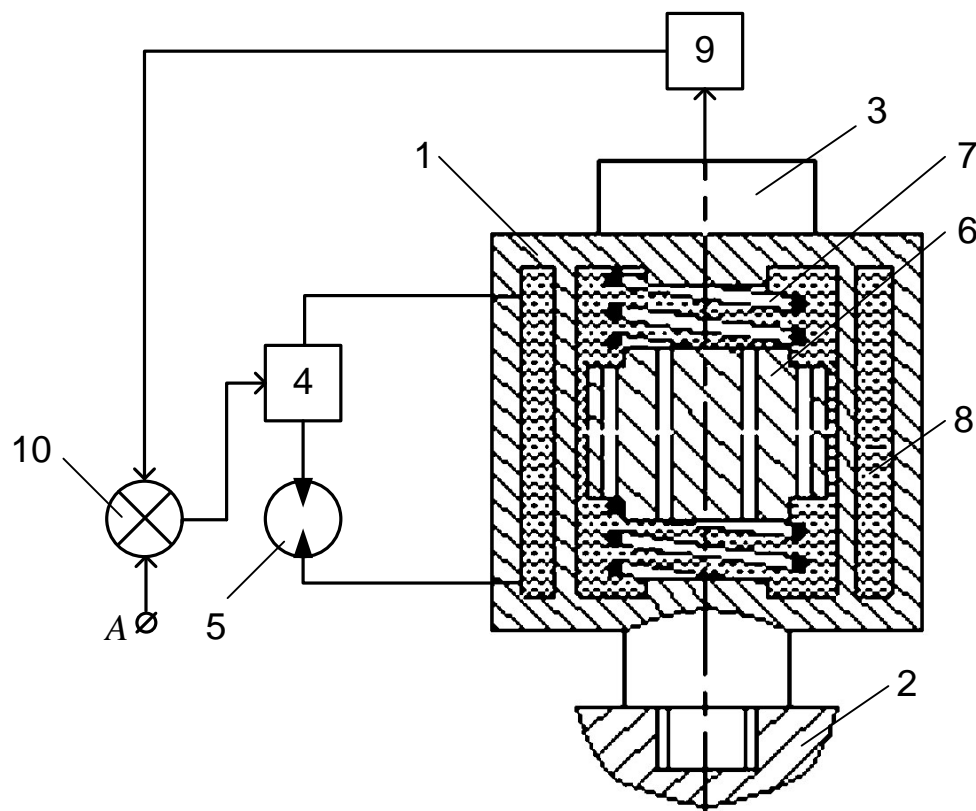


Рис. 2.8. Виброгаситель по патенту РФ № 88086

Магнитореологическая суспензия представляет собой дисперсную среду, состоящую из наночастиц (размером 10 – 50 нм) магнитных материалов и органической жидкости (масло, керосин и т. п.). Под действием магнитного поля частицы, распределенные в объеме жидкости, образуют структуры, ориентированные вдоль силовых линий поля. При этом магнитное взаимодействие между частицами приводит к значительному изменению реологических свойств среды, поскольку образованные структуры препятствуют свободному перемещению суспензии и вызывают увеличение ее вязкости до 10^5 и даже более раз.

Существует множество составов магнитореологических суспензий. Один из них, получивший довольно широкое распространение в промышленности и защищенный патентом РФ № 1154938, содержит, %: керосиновую фракцию – 24,14–47,68; аэросил – 4,57–8,8; порошок двуокиси хрома –

0,5–15,4; полимер винил-н-бутилового эфира – 1,874–3,7; α -нафтол – 0,013–0,026; олеиновую кислоту – 0,005–0,154; магнитомягкий порошок карбонильного железа – до 100. Вязкость подобной суспензии, μ , Па·с, в зависимости от напряженности внешнего магнитного поля, H , кА/м, с достаточной для инженерных расчетов точностью аппроксимируется выражением

$$\mu = 15 \cdot H - 28, \quad (3)$$

где $\mu > 30$, $H > 4$. На основе этого выражения и сконструирован виброгаситель, предложенный в [2]. Схема его устройства и размещения на крюковой подвеске крана приведена на рис. 2.9. Виброгаситель имеет корпус 1, установленный в верхней части подвески 2. Внутри корпуса расположены электромагнит 3 и магнитоареологическая суспензия 4. Имеется также груз 5, прикрепленный к корпусу 1 посредством пружин, выполненных в виде упругих стержней 6. Груз 5 (он может быть изготовлен в виде цельного кольца или отдельных частей с дросселирующими отверстиями) размещается в суспензии 4. При необходимости он может быть составным, таким, что его часть 7 может быть расположена вне корпуса 1 на наконечниках стержней 6 и установлена на резьбе. Стержни 6 также являются частью груза 5. Помимо перечисленного, виброгаситель включает в себя датчик частоты 8 и устройство управления 9 электромагнитом 3, закрепленные на корпусе 1 виброгасителя и связанные между собой. Датчик 8 измеряет частоту вынужденных колебаний подвески 2 и формирует сигнал, поступающий на устройство управления 9. Оно, в свою очередь, управляет электромагнитом 3 и вязкостью суспензии 4. Изменение вязкости влечет за собой изменение частоты собственных колебаний виброгасителя и подстройку его в резонанс.

Очевидно, что все способы адаптивного виброгашения, реализуемые в виде описанных конструктивных схем, имеют свои достоинства и недостатки и неравноценны. В связи с этим представляет интерес их сравнительный анализ, позволяющий определить, какой из них наиболее приемлем для использования в грузоподъемных кранах.

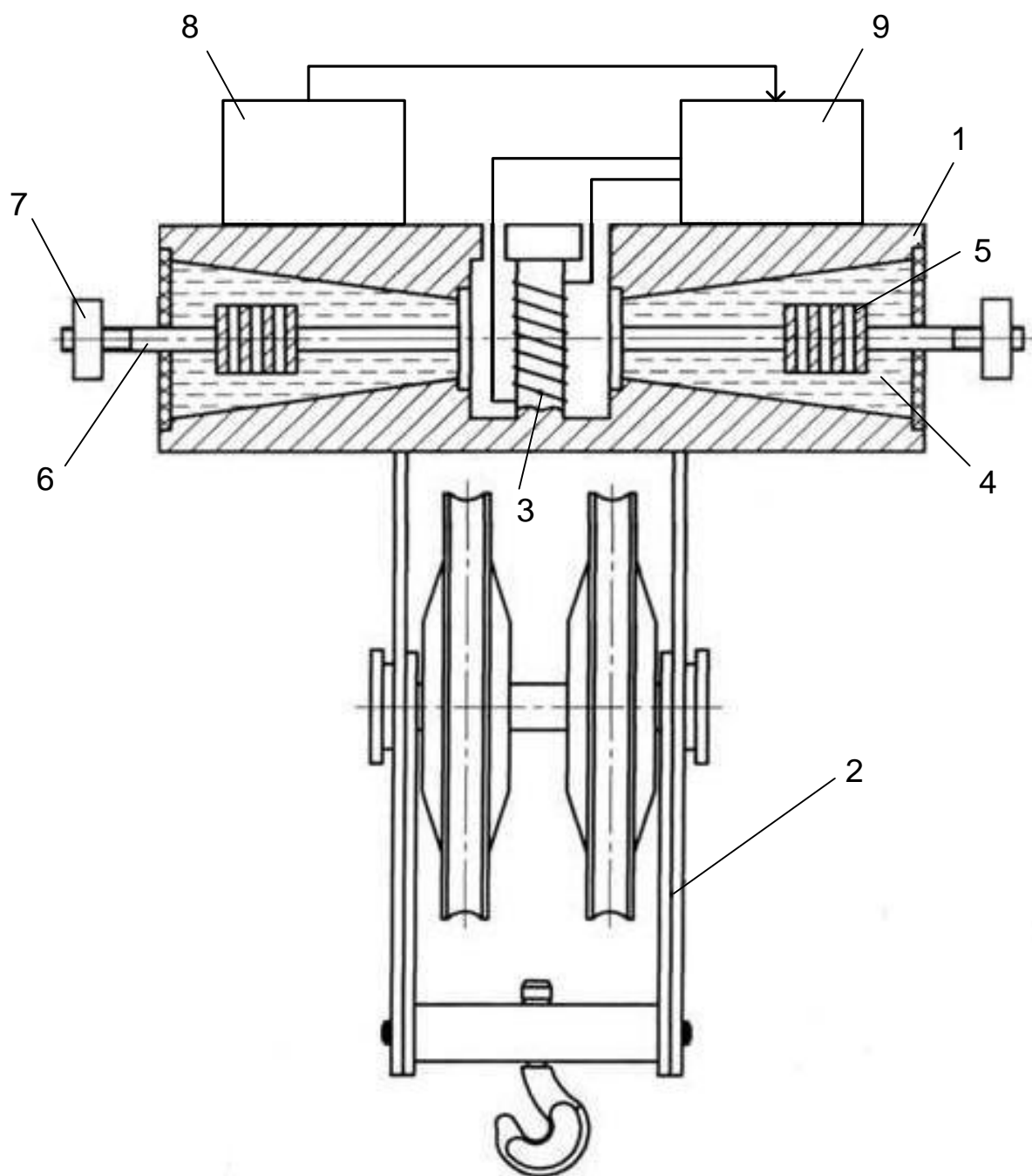


Рис. 2.9. Виброгаситель, предложенный в [2]

Сформулируем основные критерии сравнения. По-видимому, первый из них – сложность конструктивной реализации способа при современной конструкторско-технологической базе. Второй – возможные относительные габаритные размеры виброгасителя, использующего тот или иной способ, при этой же базе. Третий – относительное быстродействие (относительная продолжительность реагирования виброгасителя на изменение сигнала датчика частоты). Четвертый – надежность и долговечность. И наконец, пятый –

экономичность, необходимость применения дорогостоящих элементов, тех или иных источников энергии, их относительная потребляемая мощность и т. п.

Для проведения сравнения по перечисленным критериям был использован метод экспертных оценок [11]. Экспертам были представлены конструктивные схемы, приведенные на рис. 2.1 – 2.9, и предложено оценить их, выставя плюсы за их положительные качества и минусы за отрицательные. Уровень каждого качества нужно было оценить соответствующим числом плюсов или минусов. Результаты сравнения представлены в табл. 2.1.

Таблица 2.1

Результаты экспертной оценки способов построения
адаптивных виброгасителей

Критерий сравнения	Схема виброгасителя						
	Рис. 2.1	Рис. 2.2	Рис. 2.3	Рис. 2.4	Рис. 2.6	Рис. 2.8	Рис. 2.9
Простота конструкции	++++	+	--	++	+++	–	+++
Габаритные размеры	++	--	----	+	+++	–	+++
Быстродействие	+++	--	–	+	++	----	++
Надежность	++++	–	–	+	++	–	+++
Экономичность	++	--	+	–	++	----	+

Из таблицы, даже без специальной математической обработки полученных данных, сразу видно следующее. Наибольшее число плюсов, причем почти равное, получили виброгасители, построенные по схемам, изображенным на рис. 2.1, 2.6 и 2.9. Проще всех реализуется способ, иллюстрируемый рис. 2.1, однако и виброгасители, построенные по схемам 2.6 и 2.9, тоже не очень сложны. По быстродействию виброгаситель на рис. 2.1 также превосходит виброгасители на рис. 2.6 и 2.9, но по габаритным размерам он им уступает.

Наиболее надежным, по-видимому, будет являться виброгаситель, построенный по рис. 2.1. За ним по этому критерию следует виброгаситель по рис. 2.9 (из-за необходимости уплотнения, через которое проходят стержни 6, он несколько менее надежен), а за ним виброгаситель по рис. 2.6 (из-за использования электроуправляемого дросселя 5 его надежность еще

ниже). Таким образом, проведенный анализ выявляет три наиболее приемлемые схемы из семи. Однако какая из них все-таки наилучшая?

Очевидно, что по совокупности показателей таковой является схема по рис. 2.1. Что касается схем по рис. 2.6 и 2.9, то их преимущества друг перед другом не столь ярко выражены.

Совершенно ясно, что важную роль в процессе выбора наилучшей схемы должны играть диапазоны частот, в которых могут работать виброгасители. Если учесть, что виброгасители, построенные по схемам, изображенным на рис. 2.6 и 2.9, оба работают на основе изменения k – коэффициента вязкого трения, то какой из них лучше по диапазону частот, определить нетрудно. Полагая в формулах (1) и (2) m и c для того и другого виброгасителей одинаковыми, сравним диапазоны изменения их регулирующего параметра. Для виброгасителей по схеме 2.6 этот параметр h – ширина щели в электроуправляемом дросселе. Для виброгасителя по схеме 2.9 им является μ – вязкость магнитореологической суспензии. Однако диапазон изменения h в интересующем нас случае, как правило, не превышает 10^2 . Диапазон же изменения μ в магнитореологическом виброгасителе достигает, как отмечалось, 10^5 . Это даже при существенно различающихся F в формуле (2) влечет за собой и диапазон частот для второго виброгасителя намного более широкий, чем для первого.

Таким образом, выбирая из всех рассмотренных виброгасителей наилучший, предпочтение, вероятно, целесообразно отдавать тем, что построены по схемам, приведенным на рис. 2.1 и 2.9. Разумеется, при всех своих достоинствах и они могут быть подвергнуты доработке. Однако это вопрос перспективы их развития и применения.

2.2. Системы двухступенчатого торможения с измерением полупериода колебаний

Системы двухступенчатого торможения с измерением (прямым или косвенным) полупериода колебаний грузов, используемые сегодня в подъемно-транспортной технике, базируются на различных способах их реализации. Однако большинство из них является развитием систем, основанных на способе, описанном в [12]. Суть его заключается в том, что вторая ступень торможения подъемного механизма включается через определенное время после включения первой ступени по сигналу специального датчика замедления работы механизма.

Системы, реализующие такой способ, существенно уменьшают колебания перемещаемых грузов и снижают динамические нагрузки, действующие на кран, однако оказываются эффективными далеко не всегда. В связи с этим в работе [13] было предложено этот способ усовершенствовать – вторую ступень торможения, равную по величине момента первой ступени, включать с интервалом времени t , равным полупериоду $T/2$ продольных колебаний грузового каната.

На рис. 2.10 приведен график изменения усилий F в канате при его продольных колебаниях, возникающих в результате включения тормозных ступеней в соответствии со способом [13]. Кривая 1 – изменение усилия после включения первой ступени, кривая 2 – после включения второй. Видно, что включение второй ступени вызывает дополнительные колебания с той же амплитудой A и тем же периодом, что и включение первой ступени. Однако те и другие колебания находятся в противофазе. За счет этого они и гасятся.

На рис. 2.11 показана схема одной из систем торможения, реализующих указанный способ [14].

Система включает в себя следующее: тормоза 1 и 2 первой и второй ступеней, равных по величине тормозного момента; датчик усилия 3, уста-

новленный на грузовом канате 4; блок дифференцирования 5, вход которого соединен с выходом датчика 3; нуль-орган 6, вход которого связан с выходом блока 5; запоминающий триггер 7, вход которого связан с выходом нуль-органа 6, а выход – с цепью управления тормозом второй ступени 2. Вход А – шина обнуления триггера 7, вход В – шина включения цепи управления тормоза первой ступени 1 и датчика усилия 3 (датчик усилия подключен к цепи управления тормоза 1).

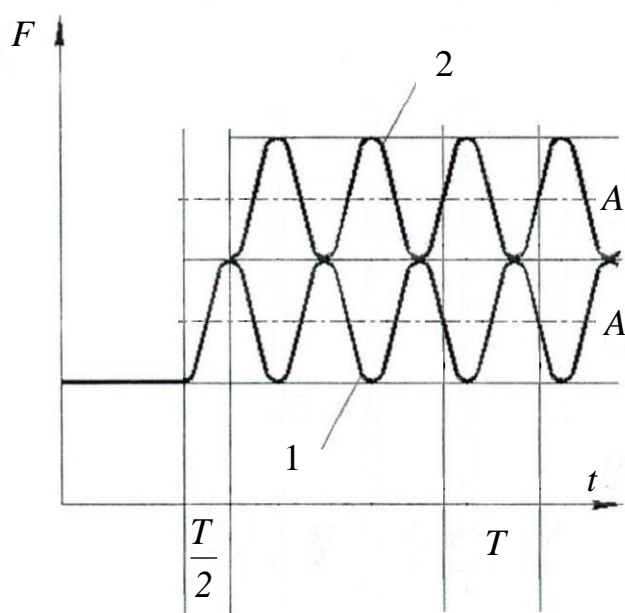


Рис. 2.10. Изменение усилий в канате при включении тормозных ступеней согласно способу по а.с. СССР № 716960

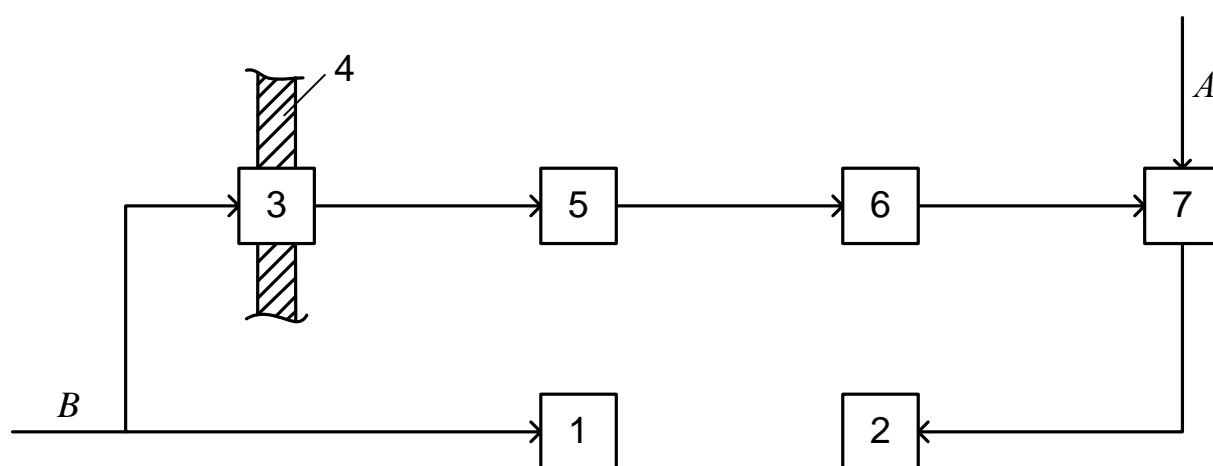


Рис. 2.11. Система торможения по патенту РФ № 68488

Перед работой подъемно-транспортной машины триггер 7 обнуляется сигналом А, на его выходе появляется сигнал «0» и тормоз второй ступени 2

выключен. Выключен тормоз первой ступени 1 и датчик 3. По окончании подъема или опускания груза, когда требуется его остановка, сигнал управления *B* включает тормоз первой ступени 1 и датчик усилия 3. Движение груза тормозится, но в канате 4 возникают продольные периодические колебания. С момента начала первого периода колебаний датчик усилия 3 выдает сигнал электрического напряжения на блок дифференцирования 5. Этот сигнал, отображая колебания усилия в канате, сначала возрастает, затем по истечении полупериода колебаний расти перестает и начинает уменьшаться. По окончании периода он вновь начинает возрастать и картина повторяется. Поступая на блок 5, сигнал от датчика 3 дифференцируется и по истечении первого полупериода колебаний усилия результат дифференцирования становится равным нулю. В результате по истечении полупериода колебаний динамического усилия в канате 4 на выходе блока 5 появляется сигнал «0». Он вызывает срабатывание нуля-органа 6, выражающееся в том, что на его выходе появляется сигнал «1». Этот сигнал переключает триггер 7 и запоминается им. При переключении триггера на его выходе появится также сигнал «1», включающий тормоз второй ступени 2. Поскольку быстроедействие блоков 5, 6 и 7 составляет микросекунды, вторая ступень торможения включается практически сразу же, как только пройдет половина периода колебаний усилия в канате с момента включения первой ступени торможения. При включении второй ступени в канате также возникают продольные периодические колебания усилия. Так как обе ступени равны по величине тормозного момента, то эти колебания будут такие, как и при включении первой ступени, но будут отставать от них по фазе на полпериода. Это приведет к тому, что колебания усилия, вызванные первой ступенью, и колебания усилия, вызванные второй ступенью, взаимно уничтожатся.

Описанная система довольно проста. Однако продолжительность полупериода колебаний при ее использовании определяется лишь приблизительно. Это обусловлено тем, что к моменту времени, когда включается

вторая ступень тормозного момента, период продольных колебаний каната еще не истек даже один раз, а потому в точности неизвестен. Длительность полупериода фиксируется первым срабатыванием нуль-органа по сигналу от элемента дифференцирования, а этот сигнал имеет большое поле рассеивания. Избавиться от этого недостатка тем не менее можно. Однако для этого способ по авторскому свидетельству № 716960 должен быть модифицирован [15] – вторую ступень торможения следует включать с задержкой времени, равной не половине периода колебаний, а половине периода плюс целое число периодов, а за время задержки измерять продолжительность периода с помощью частотомера (рис. 2.12).

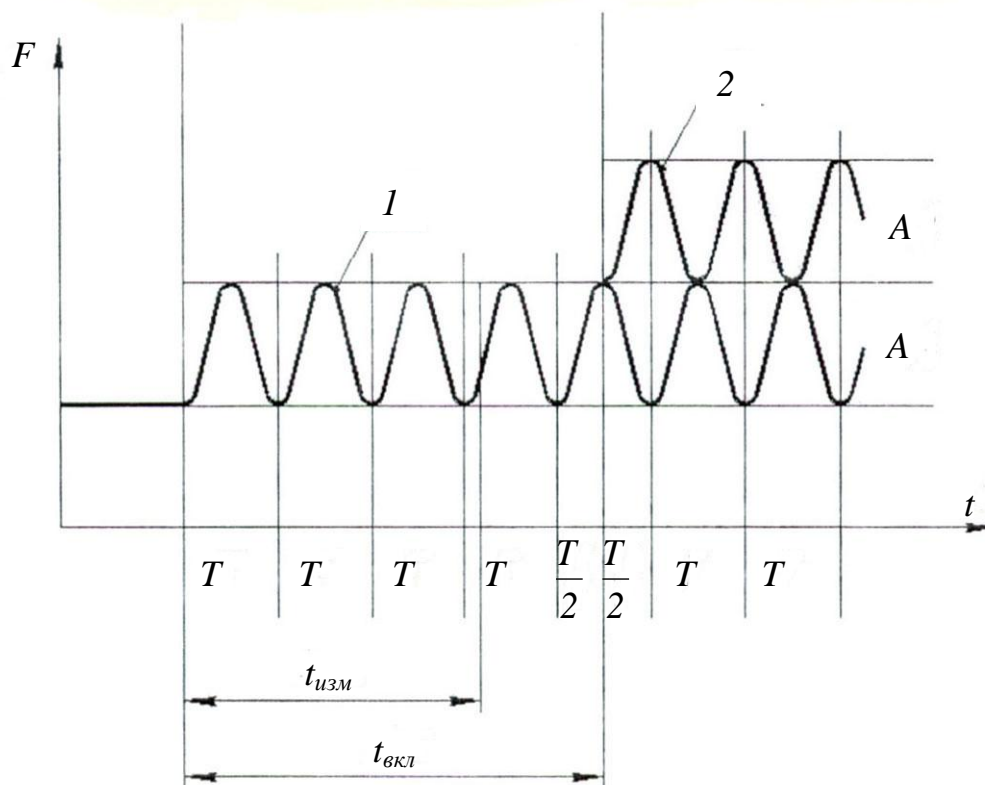


Рис. 2.12. Изменение усилий в канате при включении тормозных ступеней по патенту РФ №2342309: $t_{вкл}$ – время задержки включения второй ступени; $t_{изм}$ – время измерения периода частотомером

Схема системы торможения, реализующая модифицированный способ [16], приведена на рис. 2.13. В ней 1 – датчик усилия, установленный на грузовом канате 2; 3 – частотомер, вход которого связан с выходом датчика; 4 – реле времени, выход которого соединен с шиной ввода команды опроса

частотомера; 5 – блок деления сигналов, шина ввода делителя (сигнала, на который производится деление) которого связана с выходом частотомера 3; 6 – таймер, выход которого, так же как и выход блока деления сигналов 5, соединен с входами компаратора 7; 8 – нуль-орган, вход которого связан с выходом компаратора, а выход – с входом запоминающего триггера (*RS*-триггера) 9. Кроме того, на рис. 2.13 10 – это тормоз первой ступени, 11 – тормоз второй ступени, соединенный с выходом триггера 9; *A* – обнуляющая шина триггера 9; *B* – цепь управления тормозом 10, к которой подключены датчик 1, реле времени 4 и таймер 6, а $(K+1/2)$ – шина ввода числа $(K+1/2)$, являющегося делимым сигналом, в блок деления сигналов 5.

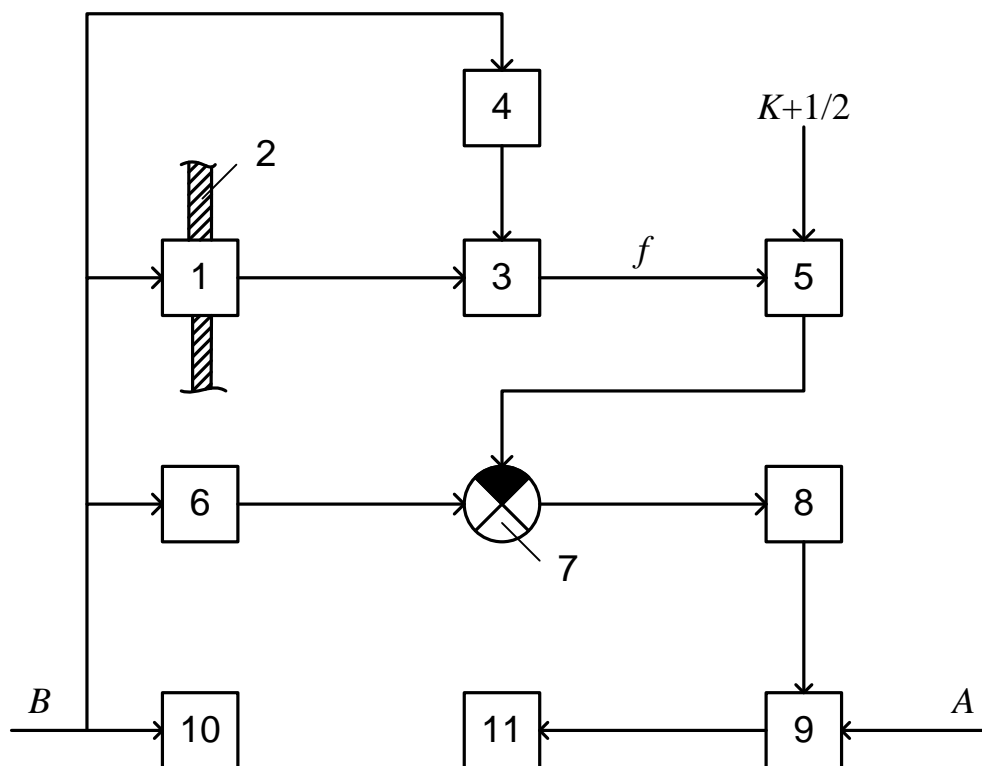


Рис. 2.13. Система торможения по патенту РФ №68489

Перед использованием системы на шину *A* подают сигнал обнуления триггера 9. Тормоза 10 и 11 выключены. Когда подъем груза завершается, подачей сигнала *B* включается тормоз первой ступени 10. Подъем груза тормозится (замедляется), но в канате начинаются продольные колебания с периодом *T*. Одновременно с включением тормоза 10 включится датчик 1, таймер 6 и реле времени 4. Таймер начнет выдавать сигнал времени, сигнал

от датчика усилия 1 поступает на частотомер 3, и последний начинает измерять частоту периодических колебаний этого сигнала. Реле времени начинает работать, отсчитывая задержку своего выходного сигнала за время T , достаточное для надежного измерения частоты продольных колебаний каната $f = 1/T$ (реле предварительно настроено на $t > MT_{max}$, где T_{max} – наибольшее возможное T или близкое к нему, а M – некоторое целое число, гарантирующее надежное измерение f). По истечении времени t реле времени 4 выдает импульс опроса на частотомер 3, а тот, в свою очередь, выдает на шину ввода делителя блока деления 5 сигнал, равный f . В блоке 5 число $K+1/2$ делится на f , а поскольку $f = 1/T$, то на его выходе появляется сигнал $(K+1/2)T$, где K – целое число, большее M , введенное в блок 5 при настройке устройства. Таким образом, на выходе блока 5 будет сигнал, отображающий время, равное $K+1/2$ периодов продольных колебаний усилия в канате. В то время как все это происходит, таймер 6 непрерывно отсчитывает время и выдает соответствующий сигнал с моментом включения тормоза 10. Поскольку этот сигнал и сигнал с выхода блока 5 подаются на компаратор, сравнивающий их (вычитающий), то как только с момента включения тормоза первой ступени 10 пройдет $K+1/2$ периодов колебаний усилия в канате, на входе нуля-органа 8 будет сигнал, равный нулю, на выходе нуля-органа появится сигнал «1», он переключит триггер 9, запомнится им, а триггер включит тормоз второй ступени. Он включится по отношению к первой ступени с задержкой, практически точно равной $K+1/2$ периода колебаний усилия в канате, т. е. со сдвигом по фазе на полпериода. Продольные колебания усилия в канате, созданные тормозами первой и второй ступеней, взаимоуничтожатся, и остановка груза после включения тормоза второй ступени произойдет почти с полной ликвидацией колебаний.

Большинство элементов рассмотренной системы достаточно технологичны и их изготовление сложности не представляет. Исключением является частотомер – устройство непростое и дорогостоящее. Наряду с этим, в данной и предыдущей системах применяется датчик усилия, устанавливаемое

схема приведена на рис. 2.14.

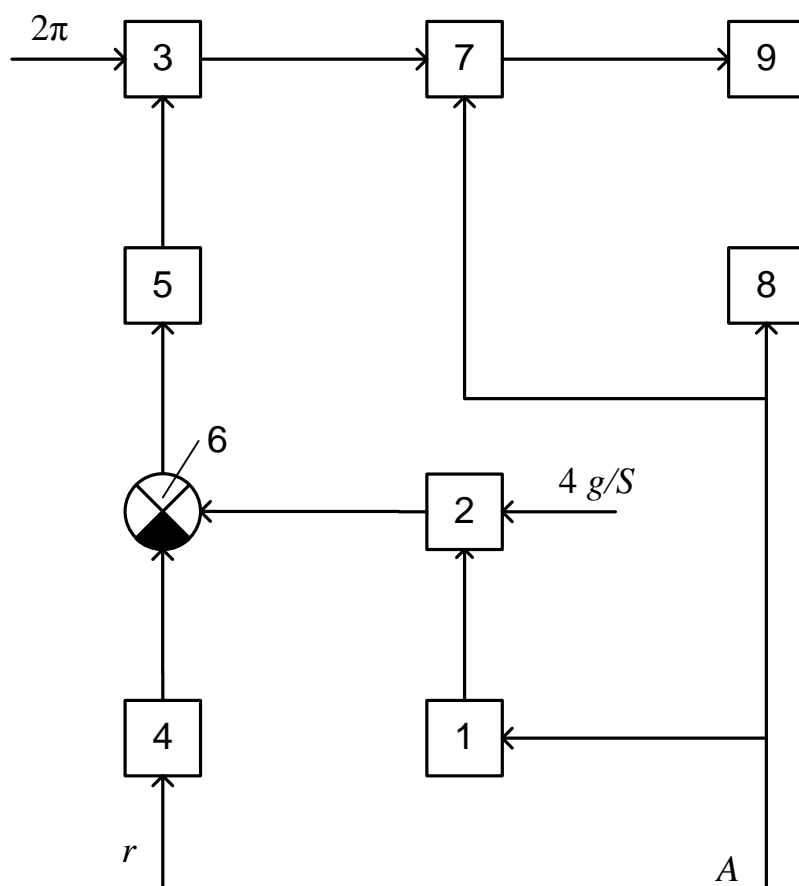


Рис. 2.14. Система торможения по патенту РФ №69055

управления тормоза 8 соединена с шиной опроса датчика 1 и с входом

включения реле времени 7. Цепь управления тормоза 9 связана с выходом реле времени 7. Выход датчика высоты 1 соединен с шиной ввода делителя в первый блок деления сигнала 2, выход делителя 2 и выход блока возведения в квадрат 4 подключены ко входам компаратора 6, выход компаратора 6 соединен с входом блока извлечения квадратного корня 5, выход блока 5 связан с шиной вывода делителя во второй блок деления сигналов 3, а выход блока 3 соединен с программирующим входом реле времени 7.

Работа системы основана на том, что, в соответствии с [3] и формулой (1), частоту колебаний груза на канате при включении тормоза можно представить как

$$f = \frac{1}{4\pi m} \sqrt{4mc - k^2},$$

где f – указанная частота; m – масса груза с крюковой подвеской; c – жесткость каната; k – коэффициент внутреннего трения в канате. Коэффициент k приблизительно можно выразить как rm , где r – некоторый (в 1/сек) множитель, при более или менее постоянной нагрузке постоянный для данного каната и определяемый экспериментально. Жесткость c может быть выражена как mg/Sl , где g – ускорение свободного падения; l – длина каната; S – безразмерный коэффициент относительной жесткости, также примерно постоянный для данного каната и тоже определяемый экспериментально. С учетом этого, через частоту f можно выразить период T продольных колебаний каната в виде

$$T = \frac{1}{f} \approx \frac{4\pi m}{\sqrt{4m \frac{mg}{Sl} - r^2 m^2}}, \quad (4)$$

или после вынесения m из-под корня и сокращения в виде

$$T \approx \frac{4\pi}{\sqrt{\frac{4g}{Sl} - r^2}}. \quad (5)$$

При включении тормоза первой ступени в канате возникнут продольные колебания с периодом T . После включения первой ступени нужно через время $T/2$ включить вторую ступень торможения. Это и осуществляет система. Перед использованием ее вначале настраивают. Для этого в первый блок деления 2 вводят сигнал, отображающий делимое $4g/S$, во второй блок деления 3 вводят сигнал, отображающий делимое 2π , а в блок возведения в квадрат 4 вводят сигнал, отображающий r . При подъеме груза краном до нужной высоты устройство срабатывает. По команде A , включающей тормоз первой ступени 8, датчик высоты подъема груза 1, работающий на вычитание, выдает сигнал $l = l_{\max} - l_T$, где l_{\max} – расстояние от точки подвеса каната до земли, а l_T – текущая высота подъема груза над землей. Таким образом, датчик 1 выдает сигнал, отображающий текущую длину грузового каната. Этот сигнал, поступая на блок 2, является делителем сигнала $4g/S$, и на выходе блока 2 появляется текущий сигнал $4g/S l$. Он поступает на компаратор 6, из него вычитается сигнал, отображающий r^2 , поступающий на компаратор от блока 4. Блок 5 извлекает из сигнала, поступающего на его вход от компаратора 6, квадратный корень и вводит результат в качестве делителя во второй блок деления 3. На выходе блока 3 появляется сигнал, отображающий $T/2$, и вводится через программирующий вход в реле времени 7. Реле начинает отсчитывать время и через время $T/2$ включает тормоз второй ступени.

Система по патенту [17] существенно проще, чем система по патенту [16], но она может быть еще больше упрощена.

Вычислим по формуле (5) величины $T/2$ для реальных значений $S \approx 0,02$ и $r \approx 15$, тогда

$$T / 2 \approx \frac{6,28}{\sqrt{\frac{1962}{l} - 225}}. \quad (6)$$

При $l = 1 - 6$ м отсюда получаются такие данные:

$l, \text{ м}$	1	2	3	4	5	6
$T/2, \text{ с}$	0,151	0,228	0,303	0,386	0,486	0,622

Но эти данные со средней погрешностью 5,2 % могут быть аппроксимированы выражением

$$T / 2 \approx 0,05 + 0,0885l \quad (7)$$

или в общем случае

$$T / 2 = a + bl .$$

То есть при включении тормоза первой ступени, как следует из изложенного, в канате возникнут продольные колебания с полупериодом, которые с точностью $\approx 5 - 6$ % можно считать близкими к

$$T / 2 = a + bl , \quad (8)$$

где a и b – некоторые действительные числа. Последнее выражение и позволяет осуществить упоминавшееся выше упрощение [18].

Рис. 2.15 иллюстрирует устройство системы, построенной на основе выражения (8). На нем 1 – датчик высоты подъема груза, работающий на вычитание (датчик длины каната); 2 – усилитель с регулируемым коэффициентом усиления (его регулировочный вход обозначен « b »); 3 – сумматор, первый (смещающий) вход которого обозначен « a », а второй (управляющий) соединен с выходом усилителя 2; 4 – программируемое реле времени, программирующий вход которого связан с выходом сумматора 3, а выход – с цепью управления тормозом второй ступени 5. Кроме того, на рисунке показан тормоз первой ступени 6 с таким же тормозным моментом, как и у тормоза 5, подключенный к своей цепи управления, и подключение реле времени 4 к этой же цепи.

Перед использованием системы усилитель 2 регулируют, устанавливая его коэффициент усиления равным « b », а на смещающий вход сумматора 3 подают сигнал, равный « a ». При подъеме краном груза датчик высоты

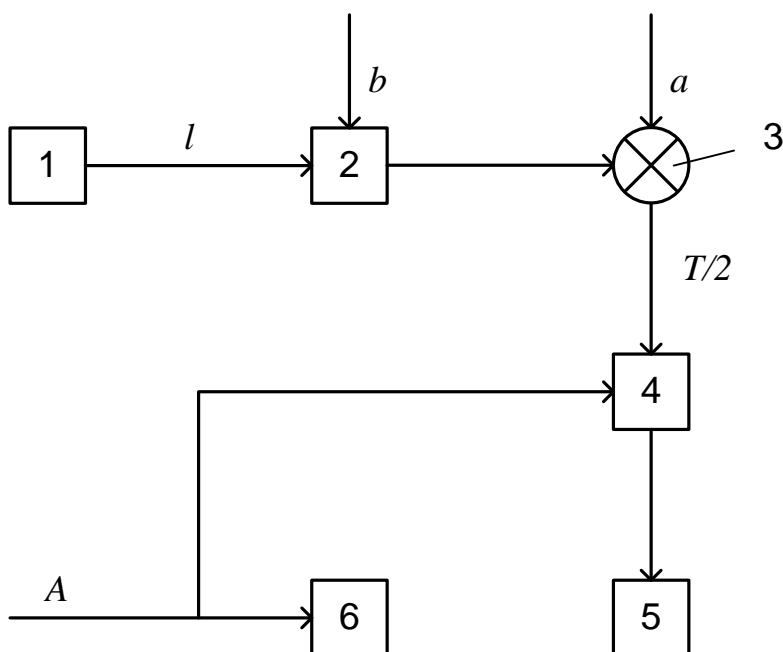


Рис. 2.15. Система торможения по патенту № 69054

подъема груза 1, работающий на вычитание, выдает сигнал, отображающий $l = l_{\max} - l_T$, где l_{\max} – расстояние от точки подвеса каната до земли, а l_T – текущая высота подъема груза над землей. Этот сигнал, поступая в усилитель 2, умножается на « b », а затем, поступая в сумматор 3, складывается с сигналом « a ». На выходе сумматора появляется сигнал, отображающий $T/2$. Он поступает на программирующий вход реле времени 4 и задает время срабатывания. Последнее же реализуется по команде включения A , подаваемой в цепь управления тормоза первой ступени 6. Когда эта команда подается, ступень 6 включается и одновременно с этим реле времени 4 начинает отсчитывать время $T/2$, которое было запрограммировано сигналом $T/2$ в момент включения ступени 6. Отсчитав это время, реле выдает сигнал включения второй тормозной ступени 5.

Система по патенту [18] – аналоговая. Вместе с тем существует и ее цифровой вариант [19]. Главной его особенностью является то, что датчик высоты подъема груза выполнен в виде кодового датчика положения. Шкала датчика положения кодирована числами, представленными в двоичной системе счисления и равными полупериоду продольных колебаний грузовой

го каната разной длины, возникающих при включении тормоза первой ступени, деленному на некоторое число τ .

На рис. 2.16,а показана схема, иллюстрирующая принцип построения шкалы кодового датчика, используемого в качестве датчика высоты подъема груза, а на рис. 2.16,б – блок-схема системы с его применением. На них 1 – развертка шкалы, изготовленной из пластика с ферромагнитным покрытием; 2 – намагниченные участки покрытия; 3 – магнитные головки, построчно считывающие кодовые комбинации со шкалы; 4 – кодовый датчик положения, содержащий свернутую в цилиндр шкалу, соединенную с барабаном, на который наматывается грузовой канат, и неподвижно закрепленные магнитные головки; 5 – двоичный счетчик импульсов, работающий на вычитание; 6 – генератор импульсов, подключенный к счетному входу счетчика 5 и формирующий импульсы с временным интервалом τ ; 7 – ключевая схема, входы которой соединены с выходами кодового датчика 4, а выходы – с шинами ввода двоичного числа в счетчик 5; 8 – логический элемент «ИЛИ – НЕ», входы которого связаны с выходами счетчика 5; 9 – логический элемент «И», первый вход которого соединен с выходом элемента «ИЛИ – НЕ» 8, второй вход соединен с включающим входом генератора 6, а выход – с цепью управления тормозом 10 второй ступени; 11 – элемент задержки, через который цепь управления тормозом 12 первой ступени, создающей тормозной момент, равный моменту тормоза 10, связана с включающим входом генератора импульсов 6. А – обозначение цепи включения устройства, соединенной с цепью управления тормозом 12 и с управляющим входом ключевой схемы 7.

Работа цифровой системы, так же как и предыдущей аналоговой, основана на использовании формулы (5). При $S \approx 0,02$, а $r \approx 15$ эта формула принимает, в частности, вид (6), откуда при $l = 1, 2, 3, 4, 5, 6$ м получаются те же величины $T/2$, что и выше. Кодовая шкала строится по числам $T/2$, соответствующим l . Для этого задается величина τ – временной интервал формирования импульсов генератором 6. Его выбирают таким, чтобы значение

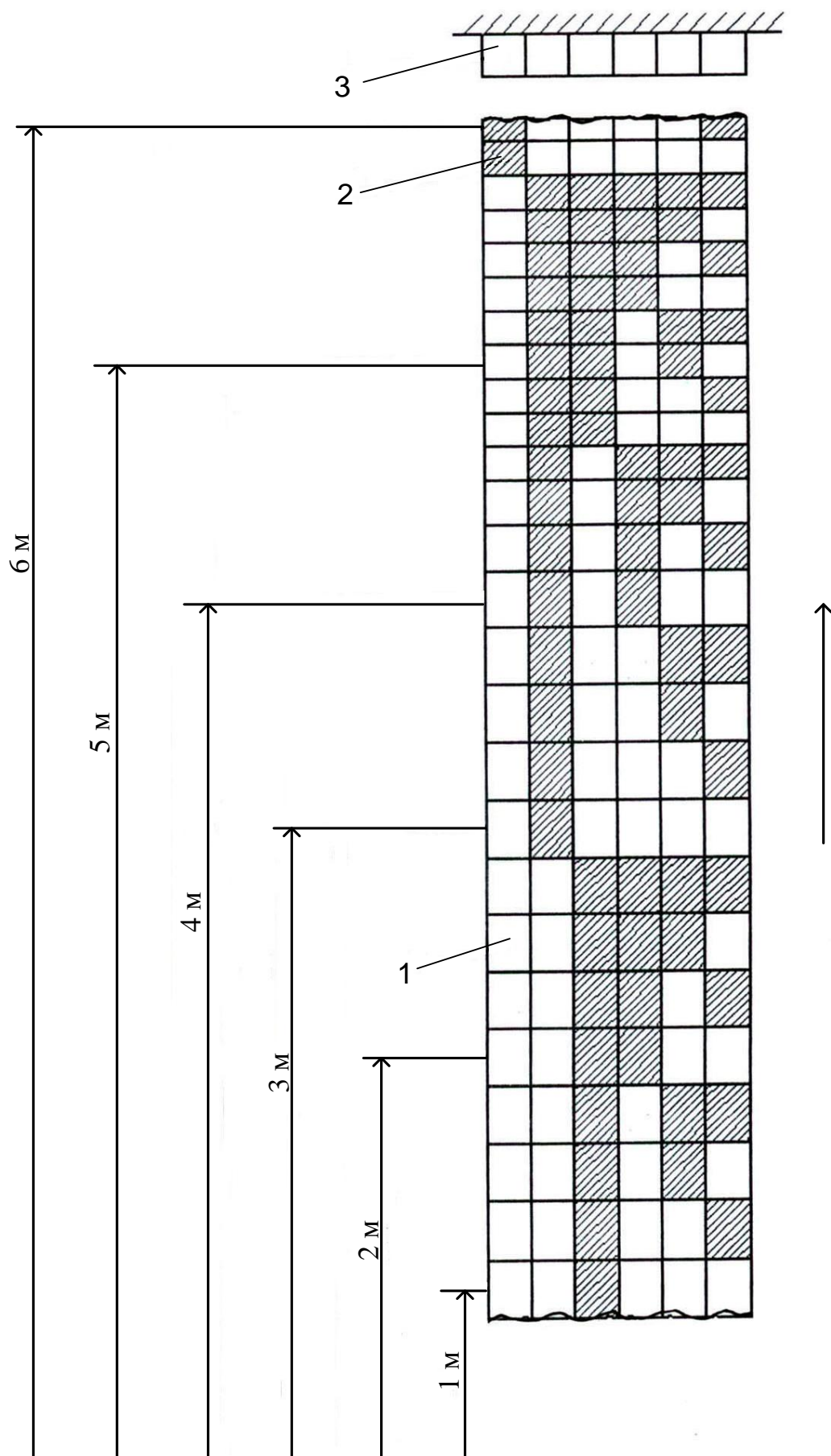


Рис. 2.16,а. Система торможения по патенту РФ № 69855
(иллюстрация принципа построения шкалы кодового датчика)

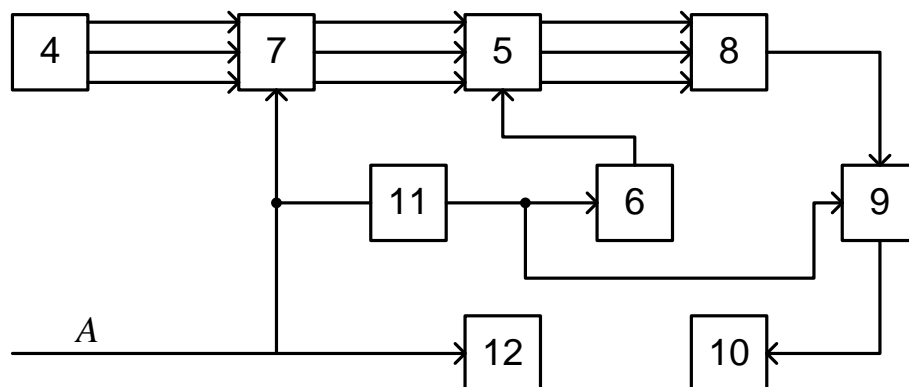


Рис. 2.16,б. Блок-схема системы торможения по патенту РФ № 69855

полупериода $T/2$ как можно точнее делилось на него нацело. Для приведенного примера этому условию отвечает $\tau = 0,0189$ с. С точностью около 2,5 % оно при $l = 1, 2, 3, 4, 5, 6$ м обеспечивает $T/2\tau$, равное соответственно 8, 12, 16, 20, 26, 33. Если теперь диапазон $T/2\tau$ от 8 до 33 представить с шагом 1, то l примерно с такой же точностью можно представить в диапазоне 1–4 м с шагом 0,25 м, в диапазоне 4–4,40 м – с шагом 0,20 м, в диапазоне 4,40–5,90 м – с шагом 0,15 м, в диапазон 5,90–6 м – с шагом 0,10 м. Если далее десятичные числа от 8 до 33 закодировать двоичными числами и поставить их в соответствие указанным значениям l в диапазоне 1–6 м, то и получится кодовая шкала, схема которой приведена на рис. 2.16,а.

По описанному принципу может быть построена любая требуемая кодовая шкала. В описываемой системе использована шкала, построенная при $\tau = 4,725 \cdot 10^{-3}$ с. При таком τ значениям $l = 1-6$ м отвечают следующие $\tau T/2$:

$l, \text{ м}$	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6
$\tau T/2$	32	40	48	56	64	73	82	91	103	116	132

Если диапазон $\tau T/2$ от 32 до 132 представить с шагом в 1, а l разделить равномерно по шагам в диапазонах $l = 1-1,5$ м, $1,5-2$ м, $2-2,5$ м и т. д., то, закодировав двоичными числами десятичные числа от 32 до 132 и поставив их в соответствие распределению l , получим шкалу, которая и применена в системе (поскольку устройство ее ясно, а изображение было бы чрезвычайно громоздким, мы ее не приводим). При работе кодового датчика положения

с такой шкалой перемещение шкалы в то или иное положение, соответствующее той или иной текущей длине грузового каната l , приводит к появлению на выходе датчика 4 двоичной кодовой комбинации, представляющей собой закодированное десятичное число $\tau T/2$. При подъеме груза эта комбинация не пропускается ключевой схемой 7. При необходимости торможения на устройство подается сигнал A (логическая «единица») и первая тормозная ступень 12 включается. В канате начинаются продольные колебания с периодом T . Одновременно с этим ключевая схема 7 пропускает кодовую комбинацию с датчика 4, соответствующую данному l , на входы счетчика импульсов 5 и вводит в него. Сразу же после этого сигнал A поступает через элемент задержки 11 на включающий вход генератора импульсов 6 (задержка производится на 2–3 мс для того, чтобы успел произойти ввод двоичного числа в счетчик) и на второй вход элемента «И» 9. Ввод кодовой комбинации в счетчик 5 приводит к тому, что эта же комбинация появляется на его выходах. В результате на выходе логического элемента «ИЛИ–НЕ» 8 появляется логический «0». Поскольку этот «0» поступает на первый вход логического элемента «И» 9, то на выходе этого элемента тоже будет «0» и вторая тормозная ступень 10 будет выключена. Генератор 6 после включения начинает выдавать импульсы. Настройка генератора такова, что импульсы выдаются с временным интервалом τ . Импульсы поступают на счетный вход счетчика импульсов 5, и поскольку счетчик вычитающий, каждый импульс будет уменьшать число, введенное в счетчик, на «1». Счетчик 5 обнулится, когда число импульсов, выданное генератором 6, станет равным числу, введенному в счетчик. Это произойдет через $T/2$ с после включения первой тормозной ступени 12. Как только счетчик 5 обнулится, на его выходах появятся одни «нули», что приведет к появлению на выходе элемента «ИЛИ–НЕ» 8 логической «единицы». На обоих входах логического элемента «И» 9 будут «единицы», на его выходе тоже появится «единица» и включится вторая тормозная ступень 10. Так как это произойдет через $T/2$ с после включения первой тормозной ступени 12, то продольные колебания каната,

вызванные включением ступени 10, окажутся в противофазе с колебаниями, вызванными включением ступени 12.

Рассмотренная цифровая система, разумеется, сложнее аналоговой. Однако она обладает намного более высокой помехозащищенностью, что особенно важно при эксплуатации кранов в зонах, насыщенных электрооборудованием. Однако и она, и предшествующие ей аналоговые варианты имеют общий недостаток: они построены исходя из условия, что масса груза, перемещаемого краном, влияет на колебания крюковой подвески с грузом незначительно. При малых массах грузов это действительно так, но с их увеличением полупериод колебаний $T/2$ в системах, предложенных в [17–19], оказывается определен все менее и менее точно. Повысить точность можно, если массы грузов принимать во внимание.

Указанное осуществлено в системе, описанной в [20]. Она разработана в предположении, что механизм влияния на $T/2$ массы груза, транспортируемого краном, аналогичен влиянию длины каната. Иначе говоря, в предположении, что

$$T / 2 = a + bl + cm, \quad (9)$$

где a , b и c – некоторые действительные числа; l – длина каната, м; m – масса груза с крюковой подвеской, кг.

Система, предложенная в [20], показана на рис. 2.17, где 1 – датчик высоты подъема груза, работающий на вычитание (датчик длины каната); 2 – усилитель с регулируемым коэффициентом усиления (его регулировочный вход обозначен « b »), подключенный к выходу датчика 1; 3 – еще один усилитель с регулируемым коэффициентом усиления (его регулирующий вход обозначен « c »); 4 – сумматор, первый (сдвигающий) вход которого обозначен « a », второй (управляющий) соединен с выходом усилителя 2, а третий (еще один управляющий) соединен с выходом усилителя 3; 5 – датчик массы поднимаемого груза; 6 – программируемое реле времени, программирующий вход которого связан с выходом сумматора 4, а выход – с цепью управления тормозом второй ступени 7. Кроме того, на рисунке показаны

тормоз первой ступени 8 с таким же тормозным моментом, как и тормоз 7, подключенный к своей цепи управления, и подключение реле времени 6 к этой цепи, а также регулируемый внешним настроечным воздействием « d » пороговый элемент 9 отключения привода подъема груза при перегрузке, вместе с усилителем 3 подключенный к датчику 5.

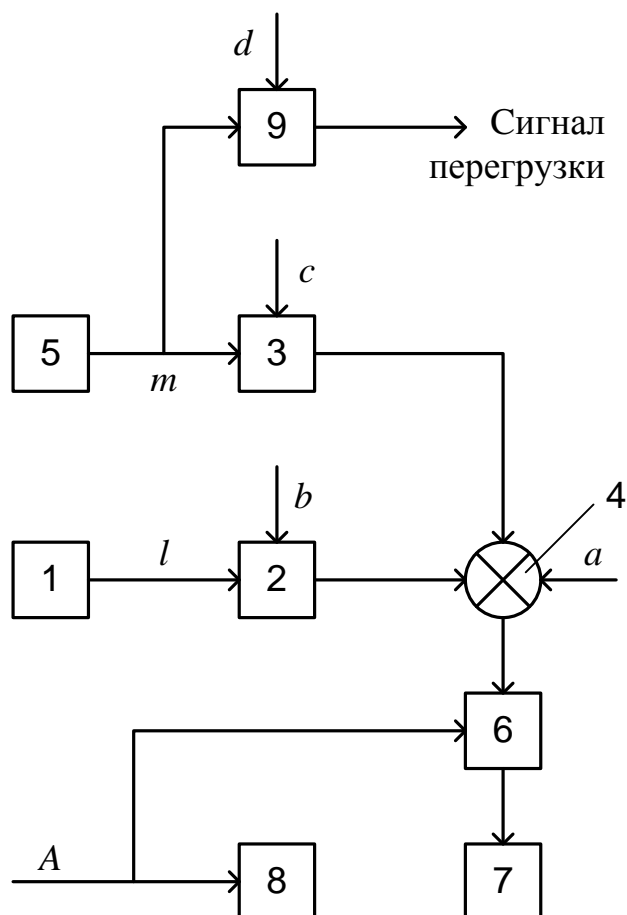


Рис. 2.17. Система торможения по патенту РФ № 87419

Перед использованием, подобно предыдущей системе, ее предварительно настраивают. Для этого усилители 2 и 3 регулируют, устанавливая их коэффициенты усиления соответственно равными « b » и « c », а на смещающий вход сумматора 4 подают сигнал, равный « a » (значения « a », « b » и « c » априорно определяют экспериментально). Пороговый элемент 9 настраивают сигналом « d » на предельно допустимое значение массы поднимаемого груза. При подъеме краном груза датчик высоты подъема груза 1, работающий на вычитание, выдает сигнал, отображающий $l = l_{max} - l_T$, где l_{max} – расстояние от точки подвеса каната до земли; l_T – текущая высота

подъема груза над землей. Таким образом, датчик 1 выдает сигнал, отображающий текущую длину грузового каната. Этот сигнал поступает в усилитель 2, умножается на « b » и, поступая в сумматор 4, складывается с сигналом « a ». Одновременно с этим датчик массы груза 5 выдает сигнал, отображающий величину поднимаемой массы. Этот сигнал, поступая в усилитель 3, умножается на « c », а затем, поступая на сумматор 4, складывается с « a » и « bl ». На выходе сумматора 4 появляется сигнал, отображающий $T/2$. Он поступает на программирующий вход реле времени 6 и задает его время срабатывания. Последнее же реализуется по команде включения «А», подаваемой в цепь управления тормоза первой ступени 8. Когда эта команда подается, ступень 8 включается и одновременно с этим реле времени 6 начинает отсчитывать время, которое было запрограммировано сигналом $T/2$ в момент включения ступени 8. Отсчитав это время, реле выдает сигнал включения второй тормозной ступени 7.

Сравним рассмотренные системы торможения, используя метод экспертных оценок, аналогично тому, как это было сделано применительно к виброгасителям в предыдущем параграфе. Исключим лишь из числа критериев «габаритные размеры», не имеющие в данном случае принципиального значения, и заменим их критерием «точность», имея в виду точность измерения полупериода колебаний.

Сведем полученные данные в табл. 2.2.

Таблица 2.2

Результаты экспертной оценки систем
двухступенчатого торможения

Критерий сравнения	Схема системы					
	Рис. 2.11	Рис. 2.13	Рис. 2.14	Рис. 2.15	Рис. 2.16	Рис. 2.17
Простота конструкции	+++	+	++	+++	—	++
Точность	--	+++	+	+	+	++
Надежность	—	—	+	++	+	+++
Быстродействие	++++	++	+++	+++	++	+++
Экономичность	+++	++	++	++	+	++

Как видно из таблицы, наивысшую оценку по точности эксперты дали системе по патенту № 68489. Это, видимо, вызвано использованием в ней

частотомера – устройства прямого измерения. Как наиболее надежную, они выделили систему по патенту № 87419. Это, вероятнее всего, есть следствие того, что они приняли во внимание наличие в большинстве современных кранов типовых и хорошо отработанных конструкций штатных датчиков высоты подъема груза и транспортируемой массы, а также то, что данная система содержит минимальное число блоков. В целом же наилучшей из всех эксперты также сочли систему по патенту № 87419.

Отметим, однако, следующее. Наилучшая из проанализированных систем двухступенчатого торможения с измерением полупериода колебаний является таковой именно по сравнению с другими системами, перечисленными в табл. 2.2. Вместе с тем она все же неидеальна. Так, в частности, она уступает системе по патенту № 68489 по точности и отстает от системы по патенту № 68488 по быстродействию. В этих направлениях она, видимо, должна еще совершенствоваться.

2.3. Устройства многоступенчатого и бесступенчатого управления приводами подъема–опускания грузов

Устройства и системы, рассмотренные выше, предназначены специально для устранения продольных колебаний каната с закрепленным на нем грузом. Между тем как дополнительную подобную функцию реализуют и устройства управления приводами грузоподъемных кранов.

Всякая система управления крановым приводом подъема – опускания, как общеизвестно, состоит из двух частей: собственно привода и устройства управления им. Первый обычно представляет собой электродвигатель с регулятором скорости, устройствами защиты двигателя и т. п. Второе – командоаппарат или контроллер. Если двигатель с регулятором образуют следящий привод, то контроллер представляет собой задатчик сигнала, изменяющегося по определенному закону, и осуществляет бесступенчатое управление. Если релейный (в смысле [21]) – то контроллер является задат-

чиком последовательности и длительности команд, подлежащих отработке приводом, и производит ступенчатое управление. Задавая приводу тот или иной режим работы, но непременно такой, чтобы при пуске привода его скорость более или менее нерезко возрастала, а перед остановкой так же нерезко снижалась, можно избежать значительных динамических нагрузок, а колебания каната либо вообще предотвратить, либо все-таки возбудить, но слабыми, быстро затухающими без принятия специальных мер.

Одним из довольно широко распространенных в настоящее время направлений построения систем управления релейными приводами является применение командоаппаратов на специальных реле: токовых, напряжения, времени [22] и др. Реле, и в частности реле времени, используются разные: маятниковые, электромагнитные, моторные.

На рис. 2.18 приведена схема системы управления релейным приводом постоянного тока с контроллером на электромагнитных реле времени.

При подготовке схемы к пуску замыкается рубильник Р, в результате чего получают питание:

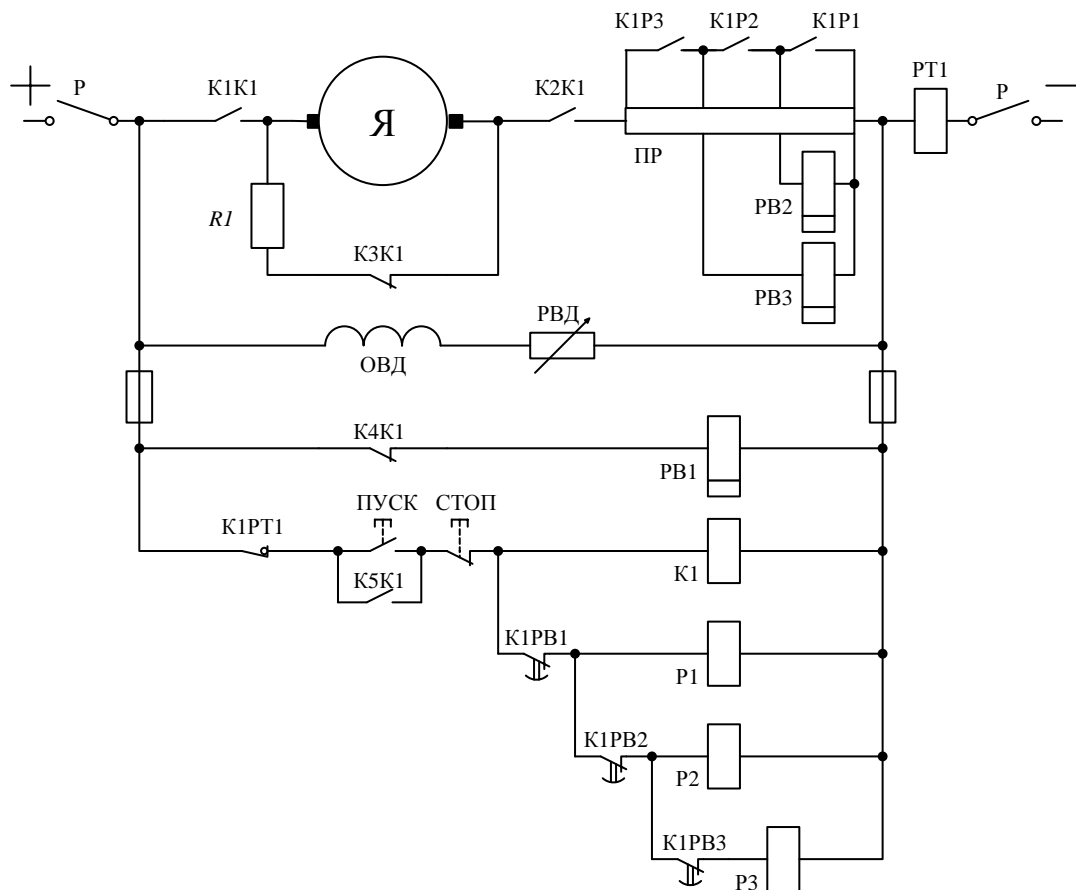
а) обмотка возбуждения двигателя ОВД (через настроечный реостат цепи возбуждения РВД);

б) катушка первого реле времени с замедлением при возврате РВ1 (через размыкающий контакт К4К1 контактора К1), которая открывает свой размыкающий контакт К1РВ1, разрывая цепь катушки промежуточного реле Р1.

После нажатия кнопки «Пуск» через размыкающий контакт К1РТ1 реле максимального тока РТ1, нажатую кнопку «Пуск», кнопку «Стоп» получает питание катушка контактора К1, которая срабатывает, и при этом:

а) закрываются замыкающие контакты К1К1 и К2К1, через пусковой реостат ПР подключающие двигатель к сети, и открывается размыкающий контакт К3К1, отключающий от цепи двигателя резистор R1; при этом катушки второго и третьего реле времени с замедлением при возврате РВ2 и РВ3 оказываются под напряжением и открывают свои размыкающие

в) открывается размыкающий контакт К4К1, который разрывает цепь катушки реле времени РВ1; при этом, поскольку реле времени РВ1 работает с замедлением при возврате (также как и РВ2 и РВ3), ее размыкающий контакт К1РВ1, открывшийся при подготовке к пуску, закроется не сразу после разрыва цепи катушки, а после некоторой выдержки времени, определяемой регулировкой реле.



Когда через установленное время контакт К1РВ1 закроется, получит питание катушка реле Р1, которое закроет свой замыкающий контакт К1Р1 главной цепи и замкнет накоротко первую ступень пускового реостата и катушку второго реле времени РВ2. Вывод ступени пускового реостата вызо-

вет бросок тока в главной цепи. Вращающий момент двигателя возрастет, а значит, возрастет и его ускорение. Замыкание накоротко катушки реле РВ2 вызовет постепенное уменьшение его магнитного потока. В результате после установленной выдержки времени закроется его размыкающий контакт К1РВ2 (который открылся после включения контактора К1) и включит в сеть катушку второго промежуточного реле Р2. Последнее сработает и закроет свой замыкающий контакт К1Р2. Теперь замыкаются накоротко вторая ступень пускового реостата и катушка РВ3. После выдержки времени замыкается размыкающий контакт К1РВ3, который включает в сеть катушку третьего промежуточного реле Р3. Последнее срабатывает, и его контакты замыкают накоротко третью ступень пускового реостата.

В рассмотренной схеме пусковой реостат имеет три ступени. При ином числе ступеней работа элементов происходит в таком же порядке.

Остановка двигателя происходит при нажатии кнопки «Стоп», которая отключает от сети катушки реле Р1, Р2, Р3 и контактора К1 (с замыканием и размыканием соответствующих контактов). Последнее приводит к открытию замыкающих контактов К1К1 и К2К1 с отключением двигателя от сети и закрытию размыкающего контакта К3К1 с подключением к двигателю резистора R1, при этом двигатель переходит в режим динамического торможения и постепенно снижает свою угловую скорость до нуля. Обмотка возбуждения ОВД в процессе торможения остается присоединенной к сети, и схема оказывается подготовленной для повторного пуска.

Схема также предусматривает максимальную и нулевую защиту двигателя. Максимальная защита осуществляется электромагнитным реле максимального тока РТ1, размыкающий контакт которого включен последовательно с кнопкой «Стоп». Нулевая защита осуществляется контактором К1, замыкающие контакты которого отключают двигатель при исчезновении напряжения в главной цепи.

В то время, как в рассмотренной схеме используется способ динамического торможения, в других подобных системах управления приводами

используются рекуперативное торможение (с отдачей энергии в сеть) и торможение противовключением. Все эти три способа в подъемно-транспортной технике реализуются весьма просто, однако имеют общий серьезный недостаток: процесс снижения скорости, осуществляемый на их основе, практически неуправляем. Тем не менее этот недостаток присущ не всем системам с релейным приводом. На рис. 2.19, например, показана схема системы с асинхронным короткозамкнутым двигателем (I) и командоаппаратом с моторным реле времени (II), в которой возможно релейное управление как разгоном, так и торможением [23]. Реле времени в ней представляет собой барабан, приводимый во вращение микродвигателем и снабженный кулачками и семью путевыми выключателями (рис. 2.20, а, б).

При подаче питания в исходном положении барабана замыкающий контакт путевого выключателя ПВ1 цепи управления реле времени и размыкающий контакт путевого выключателя ПВ2 управляющей цепи асинхронного короткозамкнутого двигателя замкнуты (первый нажат кулачком, второй находится в нормально замкнутом состоянии), а замыкающие контакты путевых выключателей ПВ3 – ПВ6 асинхронного двигателя и ПВ7 реле времени разомкнуты. При этом питание получают:

а) обмотка возбуждения микродвигателя ОВМД реле времени (через настроечный реостат цепи возбуждения РВМД);

б) реле Р2 цепи путевого выключателя ПВ1, разрывающее свой размыкающий контакт К1Р2 в цепи кнопки «Стоп» и реле Р8.

При нажатии (и отпуске) кнопки «Пуск» получает питание реле Р1 цепи управления реле времени; при этом:

а) закрывается шунтирующий замыкающий контакт К3Р1, переводя реле Р1 на самопитание;

б) закрываются замыкающие контакты К1Р1 и К2Р1, подключая микродвигатель к сети и запуская вращение барабана моторного реле вперед.

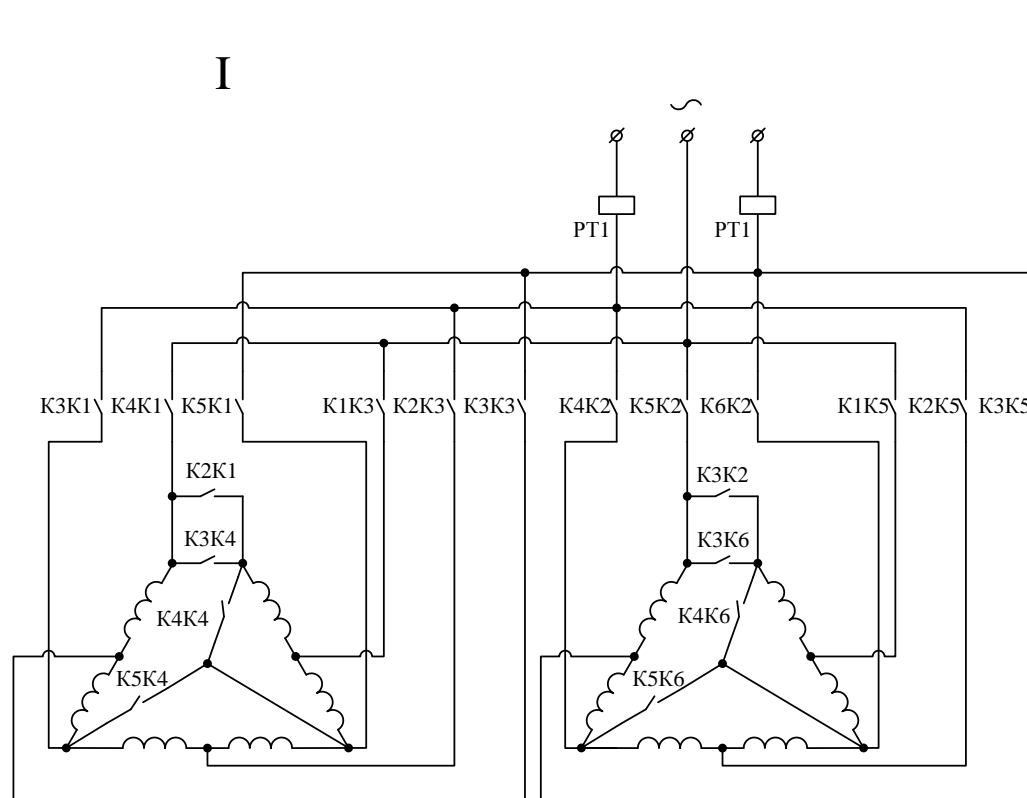
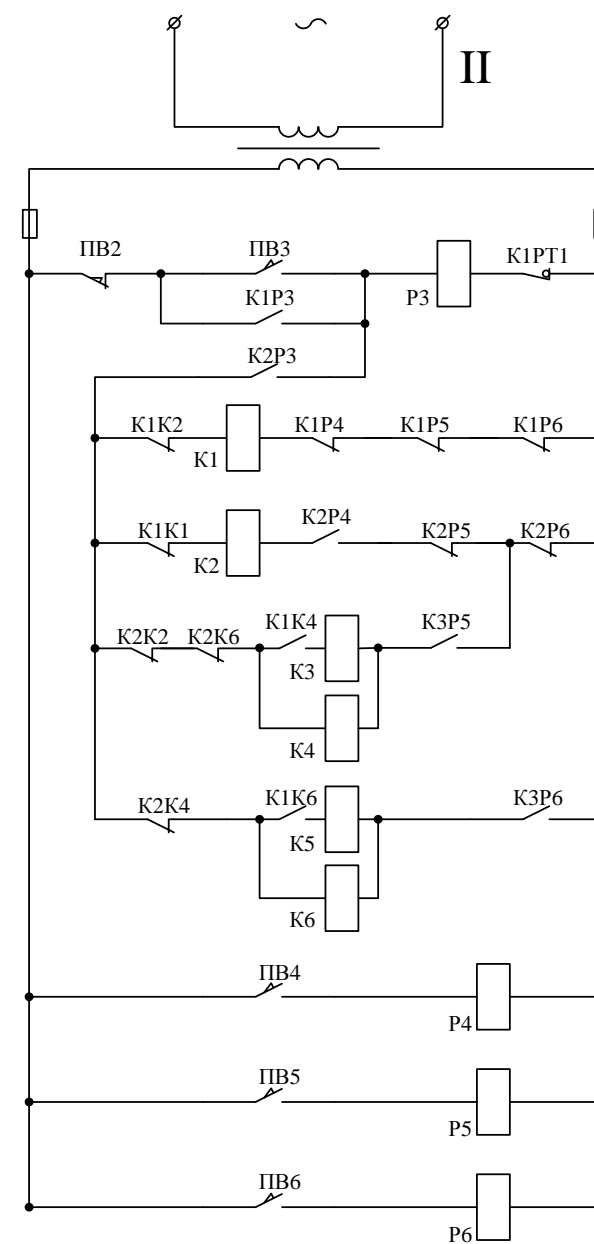


Рис. 2.19. Система с асинхронным короткозамкнутым двигателем (I) и командоаппаратом с моторным реле времени (II)



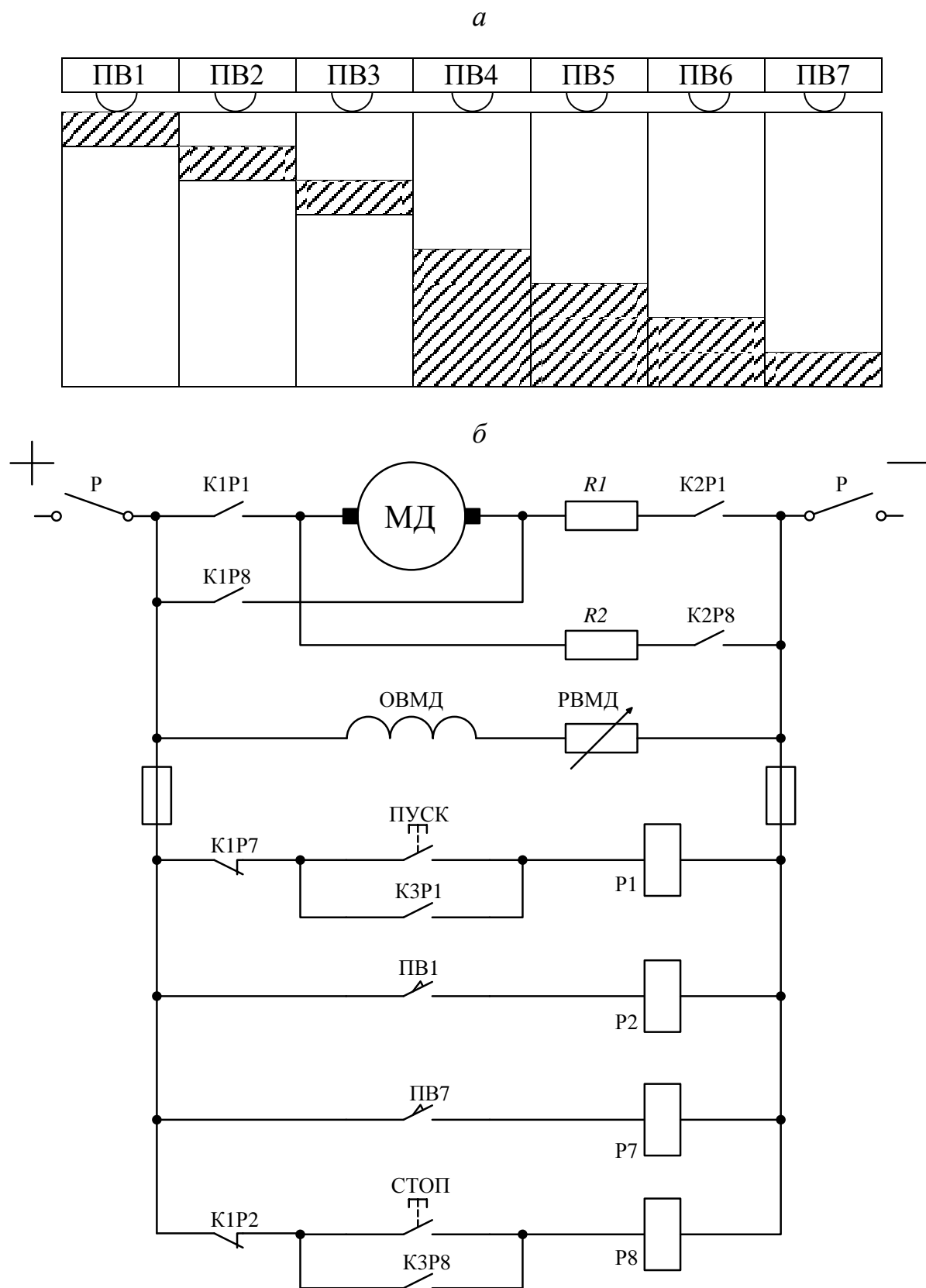


Рис. 2.20. Моторное реле времени:
a – развертка барабана с семью путевыми выключателями ПВ1 – ПВ7;
б – схема управления реле

Через определенный промежуток времени путевой выключатель ПВ1 отпускается (отключая катушку реле Р2 и закрывая размыкающий контакт К1Р2 в цепи кнопки «Стоп»), следующий кулачок барабана нажимает размыкающий контакт путевого выключателя ПВ2, относящегося уже к управляющей цепи асинхронного двигателя, размыкая его, а затем отпускает, замыкая обратно, после чего управление передается третьему кулачку, который закрывает замыкающий контакт выключателя ПВ3, подключая двухфазное питание к управляющей цепи асинхронного двигателя. Катушка реле Р3 при этом получает питание, закрывает замыкающий контакт К1Р3, шунтирующий выключатель ПВ3, переходя на самопитание, закрывает замыкающий контакт К2Р3, включая цепь контакторов К1 – К6, через контакты которых происходит непосредственное подключение и управление работой асинхронного двигателя.

Вначале возбуждается катушка контактора К1, главные замыкающие контакты которого (К2К1 – К5К1), замыкаясь в силовой цепи двигателя, присоединяют его к трехфазной сети. При этом первая обмотка статора соединяется в треугольник, и двигатель развивает минимальную скорость.

Через некоторое время после отпускания выключателя ПВ3 следующий кулачок барабана реле времени нажимает выключатель ПВ4, включается реле Р4, размыкающий контакт К1Р4 которого откроется. Это приведет к обесточиванию контактора К1, при этом отключается первая обмотка статора. С другой стороны, замыкаются размыкающий контакт К1К1 и замыкающий контакт К2Р4, включающие контактор К2, закрытие замыкающих контактов которого (К3К2 – К6К2) присоединяет к сети вторую обмотку статора, соединенную в треугольник; двигатель будет развивать вторую скорость.

Необходимо отметить, что одновременное включение контакторов К1 и К2 невозможно, так как указанные контакторы имеют кроме механической, также электрическую блокировку, осуществляемую размыкающим контактом К1К1 и замыкающим контактом К2Р4 (аналогично осуществля-

ется защита и для последующих контакторов). Схема также предусматривает максимальную и нулевую защиту двигателя. Максимальная защита осуществляется электромагнитным реле максимального тока РТ1, размыкающий контакт которого включен последовательно с выключателем ПВ3 и реле Р3, отключение которого приводит к размыканию цепи контакторов К1 – К6 и, соответственно, их контактов в силовой цепи двигателя, отключая обмотки статора от сети (здесь же через реле Р3 осуществляется и нулевая защита).

Через некоторое время пятый кулачок барабана нажимает выключатель ПВ5 – включается реле Р5 (при этом выключатель ПВ4, также как и выключатели ПВ5 – ПВ7, остается нажатым до момента отжатия его при обратном движении барабана во время торможения). В этом случае размыкающий контакт К2Р5, открываясь, отключает контактор К2, а замыкающий контакт К3Р5 включает контактор К4 и затем К3. Контактор К4 подготавливает первую обмотку статора для соединения ее в двойную звезду (через замыкание контактов К3К4 – К5К4), что в сравнении с соединением в треугольник дает увеличение скорости двигателя вдвое. После включения контактора К3 двигатель развивает третью скорость.

При дальнейшем перемещении барабана в том же направлении замкнется контакт путевого выключателя ПВ6 и включится реле Р6, размыкающий контакт которого К2Р6, открываясь, выключает контакторы К3 и К4, а замыкающий контакт К3Р6 закрывается и включает контактор К6, контакты которого в силовой цепи (К3К6 – К5К6) подготавливают включение второй обмотки статора в двойную звезду, а контакт К1К6 включает контактор К5, замыкание контактов которого разгоняет двигатель до наибольшей четвертой скорости.

Последним замыкается контакт путевого выключателя ПВ7 в цепи реле времени, подключающий реле Р7, размыкающий контакт которого К1Р7 при открытии отключает реле Р1, после чего размыкаются шунтирующий замыкающий контакт К3Р1 (дополнительно разрывая цепь кнопки

«Пуск») и замыкающие контакты цепи микродвигателя, отключая и останавливая его (т. е. дальнейшего перемещения барабана вперед не происходит, а асинхронный двигатель работает на постоянной наибольшей скорости, заданной закрытыми замыкающими контактами контакторов К5 и К6 во второй обмотке статора).

Таким образом, в конечном положении барабана реле времени замыкающие контакты путевых выключателей ПВ4 – ПВ7 и размыкающий контакт путевого выключателя ПВ2 замкнуты (первые четыре нажаты кулачком, последний находится в нормальном состоянии), а замыкающие контакты путевых выключателей ПВ1 и ПВ3 разомкнуты. Питание в цепи реле времени получает обмотка возбуждения микродвигателя ОВМД и реле Р7, в управляющей цепи асинхронного двигателя – реле Р4 – Р6 и контакторы К5 и К6 (благодаря реле Р6), в силовой цепи двигателя – вторая обмотка статора.

При нажатии (и отпускании) кнопки «Стоп» получает питание реле Р8 цепи реле времени; при этом закрывается шунтирующий замыкающий контакт КЗР8, переводя реле Р8 на самопитание, закрываются замыкающие контакты К1Р8 и К2Р8, подключая микродвигатель к сети (на обратных полюсах) и запуская вращение барабана моторного реле времени назад.

Через определенный промежуток времени путевого выключатель ПВ7 отпускается (отключая катушку реле Р7 и закрывая размыкающий контакт К1Р7 в цепи кнопки «Пуск»).

При дальнейшем перемещении барабана размыкается контакт путевого выключателя ПВ6 в управляющей цепи асинхронного двигателя, отключается реле Р6, замыкающий контакт которого КЗР6 размыкается, отключая контакторы К5 и К6, а размыкающий контакт К2Р6, замыкаясь, включает контакторы К3 и К4, переводя двигатель на третью скорость.

Через некоторое время пятый кулачок барабана отжимает выключатель ПВ5 – выключается реле Р5. В этом случае замыкающий контакт

КЗР5 отключает контактор К3 и затем К4, а размыкающий контакт К2Р5 замыкается и подключает контактор К2, переводя двигатель на вторую скорость.

Следующий кулачок барабана после некоторого промежутка времени отжимает выключатель ПВ4 – отключается реле Р4, замыкание размыкающего контакта которого (К1Р4) подключает контактор К1, а раскрытие размыкающего контакта К1К1 и замыкающего контакта К2Р4 отключает контактор К2. Двигатель переходит на первую скорость.

Через некоторое время нажимается замыкающий контакт путевого выключателя ПВ3, а затем отжимается размыкающий контакт ПВ2 (при разгоне полезную работу выполнил ПВ3, а сейчас – ПВ2), временно разрывая цепь реле Р3. Разъединение контактов К1Р3 и К2Р3 окончательно разрывает управляющую цепь асинхронного двигателя (в том числе отключая возможность самопитания реле Р3) и соответственно цепь контакторов К1 – К6, причем размыкание замыкающих контактов контактора К1 приводит к отключению двигателя от трехфазной сети с уменьшением его скорости до нуля, завершая торможение.

Последним внутри реле времени нажимается путевой выключатель ПВ1, подключая катушку реле Р2 и открывая размыкающий контакт К1Р2 в цепи кнопки «Стоп», тем самым отключая реле Р8, после чего размыкаются шунтирующий замыкающий контакт КЗР8 (дополнительно разрывая цепь кнопки «Стоп») и замыкающие контакты К1Р8 и К2Р8 цепи микродвигателя, отключая и останавливая его (т. е. дальнейшего перемещения барабана назад не происходит (он вернулся в то же состояние, что и перед разгоном), а асинхронный двигатель остановлен и отключен от сети).

Таким образом, последовательность нажатия путевых выключателей и изменение состояний цепей управления и асинхронного двигателя при описанном четырехступенчатом торможении зеркально отражают аналогичные процессы при разгоне.

Сложность и громоздкость контроллера с моторным реле времени

из приведенного примера очевидна, поэтому, несмотря на значительную коммутационную мощность, высокую помехоустойчивость и другие достоинства, в последние годы такие все больше и больше из практики управления релейными приводами вытесняются контроллерами на программируемых логических матрицах [24]. Командоаппараты на таких матрицах могут выполнять функции и моторного реле времени, и в определенной степени сопряженной с ним электросхемы (такой, как II на рис. 2.19). При этом они оказываются существенно долговечнее и экономичнее, имеют меньшую массу и, кроме того, в отличие от устройств с «жесткой» логикой, обладают большей универсальностью и легко могут быть адаптированы по отношению к конкретным условиям эксплуатации.

Контроллер с программируемой логической матрицей обычно включает в себя собственно матрицу и блок чтения информации с нее, построенный на реверсивном регистре сдвига (рис. 2.21).

Матрица 1 состоит из входных 2 и выходных 3 шин, соединяемых диодами. Соединения шин через диоды производятся при программировании матрицы (точки соединений при программировании на рисунке обозначены знаками *). К выходным шинам матрицы подключены релейные исполнительные элементы 4 (электронные или электромагнитные реле, обозначенные на рисунке, например, *a, b, c, d*). К входным шинам матрицы подключен своими выходами реверсивный регистр сдвига 5.

Важным блоком контроллера является генератор импульсов 6, выход которого соединен с первыми входами первого 7 и второго 8 логических элементов «И» (ключей). Выход ключа 7 связан с прямым входом первого логического элемента «НЕТ» 9 (элемента запрета), а выход ключа 8 – с прямым входом второго логического элемента «НЕТ» 10. Выходы элементов «НЕТ» подключены к шинам сдвига регистра 5 (выход элемента 9 – к шине 11 прямого сдвига, выход элемента 10 – к шине 12 обратного сдвига).

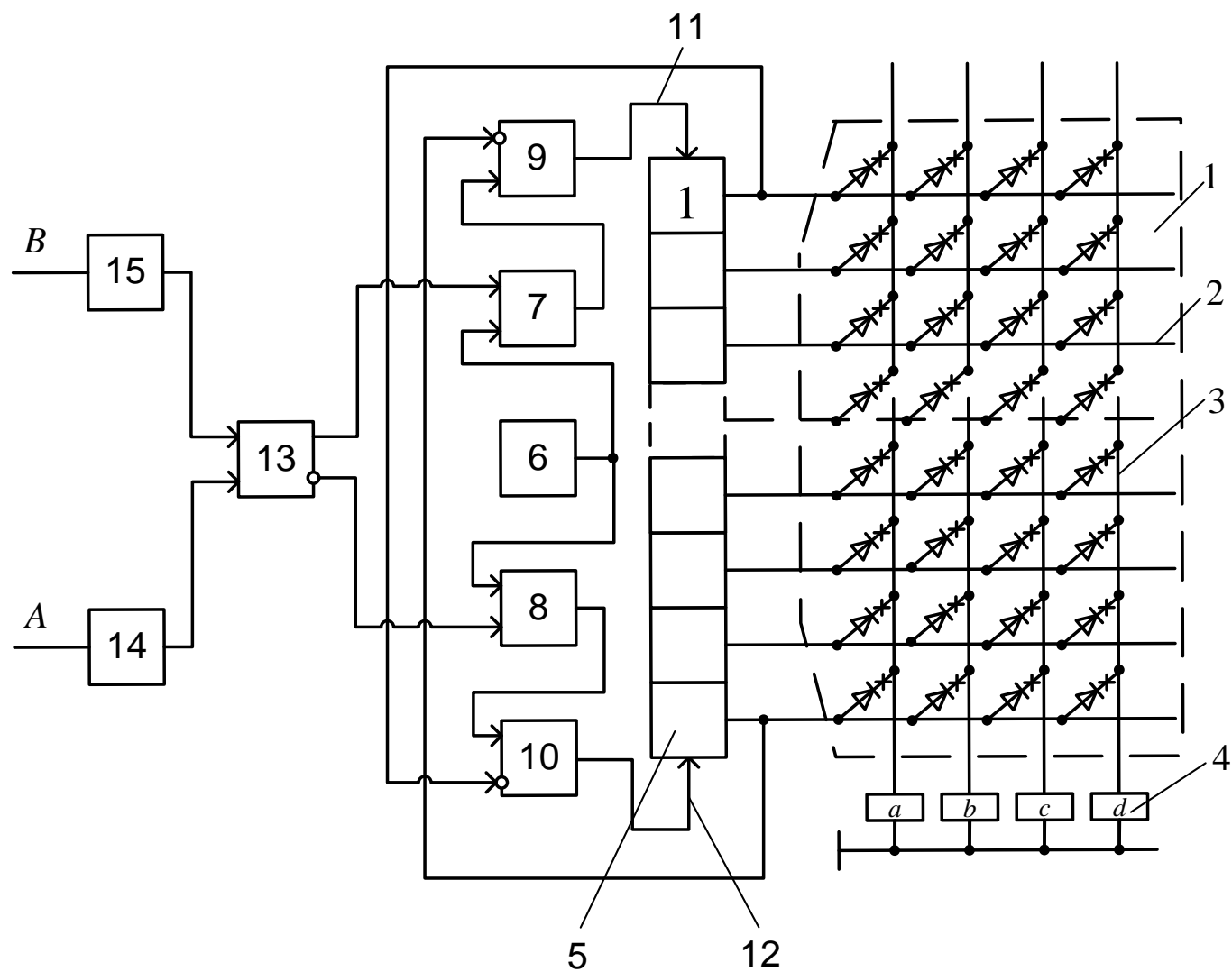


Рис. 2.21. Контроллер с программируемой логической матрицей

Кроме того, контроллер включает в себя RS-триггер 13, прямой выход которого соединен со вторым входом ключа 7, а инверсный выход – со вторым входом ключа 8. Первый вход триггера 13 подключен к шине пуска контроллера *A* (для надежности подключение может быть выполнено через формирователь импульса 14), а второй вход триггера – к шине останова *B* (подключение также может быть выполнено через аналогичный формирователь 15). В дополнение к этому, выход последней ячейки регистра 5 соединен с инвертирующим входом логического элемента 9, а выход первой ячейки регистра – с инвертирующим входом логического элемента 10.

При наладке контроллера в первую ячейку регистра сдвига 5 записывают логическую «единицу». В зависимости от конструкции релейного привода, для управления которым должен быть применен контроллер, определяют, каков должен быть закон разгона и торможения привода и каковы должны быть для его реализации последовательности комбинаций включаемых исполнительных элементов 4 контроллера во времени. (Например, для разгона привода исполнительные элементы *a*, *b*, *c*, *d* должны последовательно включаться в порядке: *a*, *b*; *a*, *c*; *a*, *b*, *c*; *b*, *c*, *d* и т. п. Для торможения же привода эта последовательность должна быть обратной). В соответствии с требуемой последовательностью комбинаций включаемых исполнительных элементов 4 матрица 1 контроллера программируется.

При использовании контроллера включают генератор импульсов 6. (Частота выдаваемых им импульсов соответствует нужному темпу разгона или торможения управляемого привода). Одновременно с этим (или чуть раньше) подают сигнал «пуск» на шину *A*. На прямом выходе триггера 13 оказывается «1», на инверсном – «0». Логический элемент «И» 7 оказывается включенным, а элемент «И» 8 – выключенным. Импульсы от генератора проходят через элемент 7, затем через элемент 9 на шину сдвига 11 регистра, и «1» в регистре начинает в нем последовательно сдвигаться, подавая напряжение на первую, вторую, третью и так далее входные шины 2 матрицы 1. В соответствии с заданной программой будет появляться

напряжение на выходных шинах 3 матрицы и последовательно в запрограммированных комбинациях будут включаться исполнительные элементы 4 контроллера, подавая на управляемый привод соответствующие сигналы и разгоняя его по заданному закону. Когда «1» в регистре 5 дойдет до последней ячейки, на инвертирующий вход логического элемента «НЕТ» 9 также поступит «1», и этот элемент перестанет пропускать импульсы от генератора 6 на шину 11. Сдвиг в регистре прекратится, и исполнительные элементы 4 контроллера окажутся в состояниях, соответствующих окончанию разгона управляемого привода, т.е. его рабочему ходу. Когда потребуются привод остановить, предварительно выполнив его торможение, на шину В контроллера должен быть подан сигнал «Стоп». Триггер 13 переключится, на его прямом выходе появится сигнал «0», а на инверсном – «1». Импульсы от генератора 6 начнут проходить через ключ 8 и элемент «НЕТ» 10 на шину обратного сдвига 12 регистра 5. Комбинации включенных исполнительных элементов 4 последуют во времени в обратном порядке, а сигналы от них, поступая на управляемый контроллером привод, начнут его скорость снижать. Так будет продолжаться до тех пор, пока логическая «1» в регистре не окажется в первой ячейке. Это приведет к подаче сигнала запрета на инвертирующий вход логического элемента 10 и прекращению поступления импульсов на шину 12. Если в первой строке матрицы запрограммирована комбинация включения исполнительных элементов 4, соответствующая останову привода, то он и остановится. После требуемого выстоя привод может быть снова введен в режим разгона сигналом «Пуск» на шину А – и все аналогичным образом повторится.

Контроллер с программируемой логической матрицей и регистром сдвига может быть изготовлен на интегральных микросхемах. При таком исполнении он получается довольно компактным и обладает высокой ремонтопригодностью. В случае необходимости он может использоваться для управления не только релейными, но и следящими приводами. Для этого в качестве его исполнительных элементов применяют не электрон-

ные или электромагнитные реле и контакторы, а усилители с регулируемыми коэффициентами усиления. Выходы усилителей подключаются к сумматору напряжений, а выход сумматора – к управляющему входу привода. Настройка контроллера осуществляется путем программирования матрицы и настройки коэффициентов усиления усилителей. Производится это так, чтобы при работе регистра и последовательном перемещении «1» по выходам регистра на выходе сумматора напряжение при необходимости разгона привода возрастало, а при необходимости торможения – уменьшалось.

Контроллеры с программируемой логической матрицей и регистром сдвига для их использования совместно с релейным приводом обычно строятся в расчете на получение до восьми ступеней разгона и торможения привода. Это обусловлено производимой в настоящее время номенклатурой таких приводов. И хотя они иногда, как отмечалось, при соответствующих дополнениях схемы применяются и для управления следящими приводами, это все же нецелесообразно. Причина ясна: в подобных случаях способность следящих приводов к существенно более плавному разгону и торможению реализуется далеко не полностью. Разумеется, этого в определенной мере можно избежать путем увеличения числа ячеек в регистре и соответствующего увеличения числа других элементов контроллера (регулируемых усилителей и пр.). Однако это вряд ли рационально. В связи с этим для управления следящими приводами чаще, чем рассмотренные выше, применяются контроллеры, близкие к ним по принципу действия, но несколько иные. Вместо регистра в них используется реверсивный двоичный счетчик импульсов (рис. 2.22). Выходными блоками таких контроллеров являются преобразователь «код–напряжение» (ЦАП) 1, формирующий сигнал C , подаваемый на привод, и ограничители этого сигнала – пороговый элемент 2, настраиваемый сигналом 3 на допустимый максимум сигнала C , и нуль-орган 4. Выходы реверсивного счетчика 5 подключаются ко входам ЦАП. Остальные элементы контроллера со счетчиком те же, что и контроллера с программируемой матрицей, с той лишь разницей,

что выход элемента запрета 9 соединен с суммирующим входом 11 счетчика 5, выход элемента 10 – с вычитающим входом 12 счетчика, с инвертирующим входом логического элемента «НЕТ» 9 соединен выход порогового элемента 2, а с инвертирующим входом элемента «НЕТ» 10 – выход нуль-органа 4.

Работает контроллер со счетчиком практически так же, как с регистром сдвига. Когда генератор импульсов 6 включен и на контроллер подан сигнал *A* «Пуск», ключ 7 и элемент «НЕТ» 9 пропускают импульс на суммирующий вход 11 счетчика 5. На выходе счетчика число, представленное в двоичной системе счисления, постепенно возрастает, преобразователем 1 оно преобразуется в напряжение, которое, соответственно, тоже возрастает, и привод, управляемый контроллером, разгоняется сигналом *C*. Разгон продолжается до момента срабатывания порогового элемента 2. В этот момент импульсы от генератора 6 на суммирующий вход 11 счетчика 5 проходить перестают и сигнал *C* расти перестает. Разгон привода прекращается, и идет рабочий ход. Для его прекращения и торможения следящего привода, управляемого сигналом *C*, на вход *B* контроллера подается сигнал «Стоп», импульсы от генератора 6 начинают поступать через ключ 8 и элемент «НЕТ» на вычитающий вход 12 счетчика, сигнал *C* начинает уменьшаться, скорость работы привода снижается, и, когда сигнал *C* станет равен нулю, нуль-орган 4 переключит элемент «НЕТ» 10 и счетчик 5 остановит. Таким образом, рассматриваемый контроллер, подобно предыдущим, заставляет работать привод по закону «разгон–рабочий ход–торможение». Однако поскольку счетчик импульсов в нем может иметь весьма большое количество состояний (при восьми разрядах, например, $2^8 = 256$ состояний), то при соответствующем числе разрядов ЦАП сигнал *C* оказывается изменяемым практически бесступенчато.

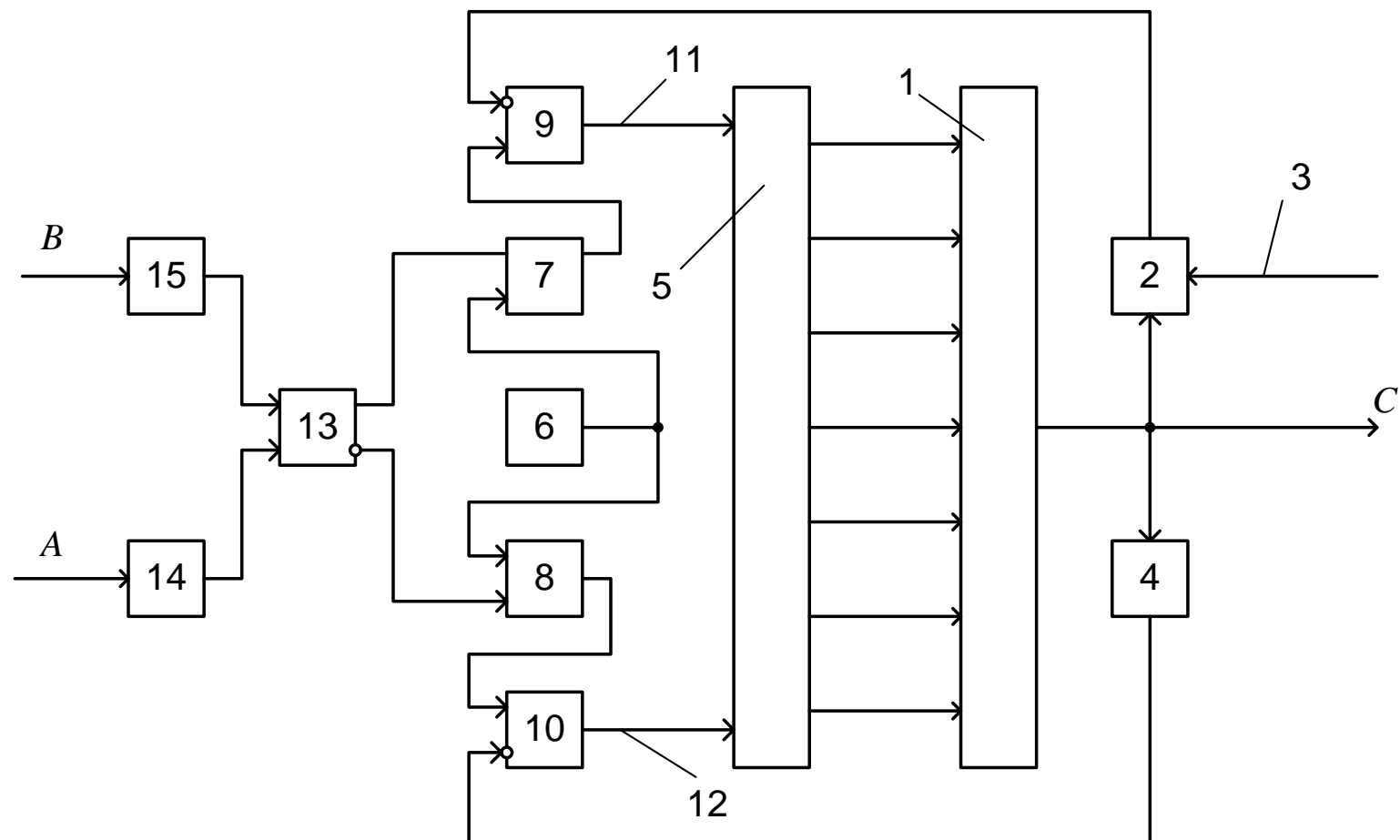


Рис. 2.22. Контроллер с реверсивным счетчиком импульсов

При управлении следящим приводом с помощью контроллера с реверсивным счетчиком закон «разгон–рабочий ход–торможение» привода с точностью до дискретности счетчика будет выглядеть так, как показано на рис. 2.23, где $t_{разг}$, $t_{раб}$, $t_{торм}$ – длительности элементов закона, $C_{раб}$ – величина C , соответствующая скорости рабочего хода $V_{раб}$. Параметры закона можно задавать разными. Если при разгоне привода частоту работы генератора ω принять меньшей, чем при торможении, то $t_{торм}$ будет меньше $t_{разг}$. И наоборот. Меняя настройку порогового элемента 2, можно задавать ту или иную величину $V_{раб}$. Это делает эксплуатационные возможности контроллера весьма широкими, хотя и не лишает его и некоторых недостатков. Так, в частности, при работе контроллера в некоторых точках изменения сигнала C (кривая 1) имеют место значительные скачки производной (ускорения) C (кривая 2).

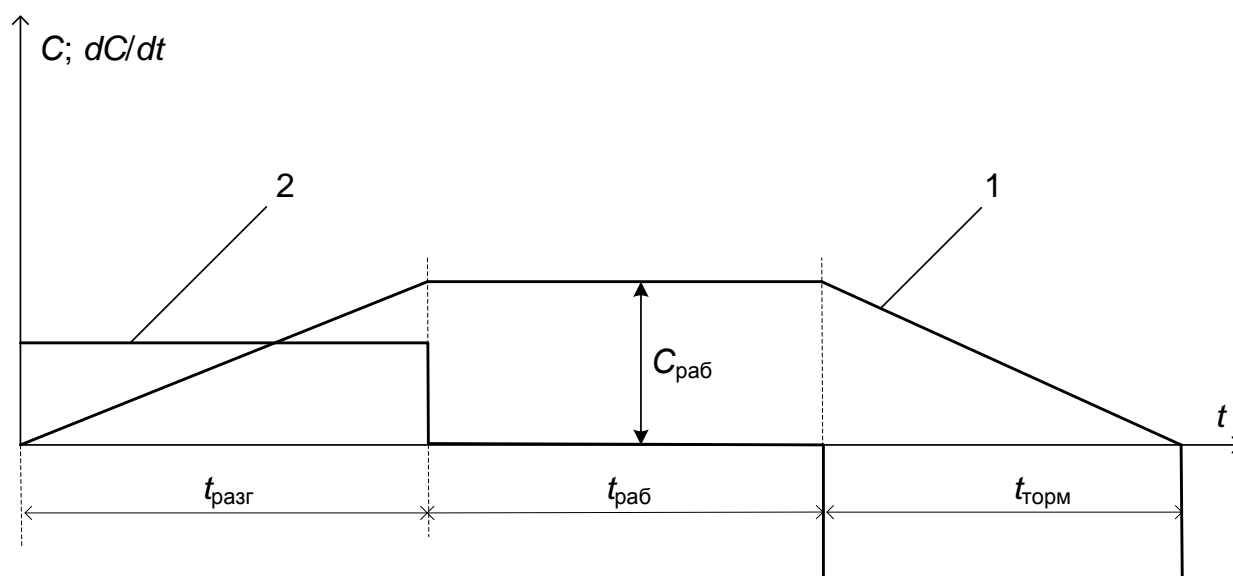


Рис. 2.23. Изменение сигнала C и его производной при работе контроллера со счетчиком импульсов

Если в качестве ЦАП применить преобразователь с последовательным делителем напряжения, то эти скачки можно несколько сгладить, но все же не ликвидировать полностью.

3. ИССЛЕДОВАНИЕ ВЕРТИКАЛЬНЫХ КОЛЕБАНИЙ ГРУЗА НА ГИБКОЙ ПОДВЕСКЕ ПРИ ТОРМОЖЕНИИ

3.1. Задачи и методика исследования

Рассмотрев существующие средства устранения вертикальных колебаний грузов при их подъеме и опускании, естественно поставить задачу дальнейшего совершенствования наиболее перспективных из них. Однако для этого необходимо провести дополнительное исследование. В частности, исследование колебаний груза при торможении механизма подъема груза при подъеме и опускании последнего. Целью такого исследования, очевидно, должно являться определение наиболее адекватного вида зависимости частоты указанных колебаний от длины каната и массы груза, поскольку, если такая зависимость будет найдена, она даст возможность повысить точность подстройки в резонанс виброгасителей, описанных в п. 2.1, и согласованность включения ступеней систем двухступенчатого торможения, описанных в п. 2.2.

Разумеется, для проведения исследования необходима специальная экспериментальная установка, ориентированная на реализацию вполне конкретной методики. Кроме того, нужно сразу же решить, какой математический и вычислительный аппарат потребуется для обработки данных, получаемых в ходе эксперимента, так как возможности и специфические особенности этого аппарата определяют и вид представления данных, подлежащих обработке. Учитывая изложенное, для проведения эксперимента была предложена следующая методика.

1. В ходе эксперимента с помощью некоторой грузоподъемной машины, оснащенной быстродействующим электроуправляемым тормозом, нужно поднимать определенный груз до определенной высоты и на этой высоте резко затормаживать.

2. Колебания, возникающие в результате затормаживания, следует записывать, используя некоторый датчик, измеряющий либо непосредственно перемещения груза в процессе колебаний, либо функционально связанную с ними величину.

3. Высоту, на которой производится затормаживание и запись колебаний, следует изменять с некоторым шагом, позволяющим достоверно различать разницу в параметрах колебаний.

4. Опыты, выполняемые по пунктам 1–3, следует повторить, изменяя массу груза с шагом, также обеспечивающим достоверное различение разницы в параметрах колебаний.

5. Опыты, выполняемые по пунктам 1–4, следует повторить, не поднимая, а опуская грузы определенной массы до определенной высоты.

6. Обработку данных о частотах колебаний при соответствующих высотах подъема и опускания грузов соответствующей массы следует производить с помощью компьютерной системы, позволяющей отыскивать параметры математических моделей различных видов, описывающих зависимость частоты от длины грузового каната и массы груза; при этом используемая система должна обеспечивать проверку каждой модели на адекватность.

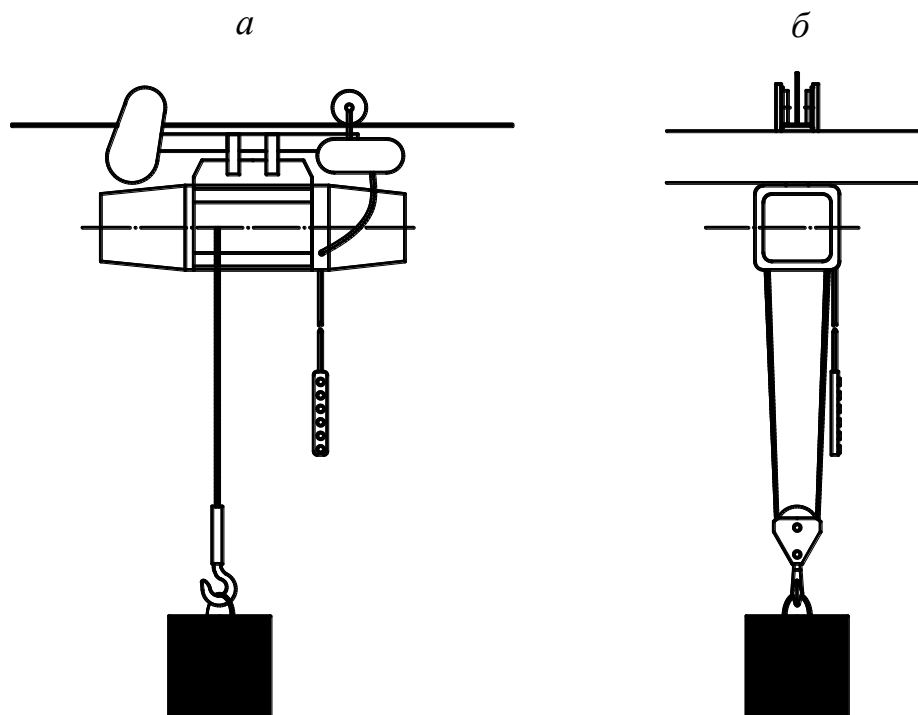
Покажем далее, как описанная методика была применена на практике.

3.2. Экспериментальная установка

Для решения поставленных задач в соответствии с предложенной методикой была разработана экспериментальная установка [25], состоящая из следующих компонентов: электрическая канатная таль марки Э025-51120-31У2; виброаппаратура ВИ6-6ТН с датчиком виброускорения; аппаратно-программный комплекс, содержащий специальный аналого-цифровой преобразователь (АЦП) и типовой компьютер РС.

Таль (рис. 3.1, *а, б, в*) имеет грузоподъемность 0,5 т, обеспечивает скорость подъема груза 0,13 м/с и питается переменным напряжением 380 В / 50 Гц. Она состоит из механизма подъема, механизма передвижения и крюковой подвески, содержащей крюк со шкивом, подвешенный на двухстропном многожильном стальном канате диаметром 5 мм типа ЛК-0 (ГОСТ 3063-80). Механизм подъема – основная часть тали, задействованная в исследованиях. Он включает в себя корпус, барабан с винтовой нарезкой, асинхронный электродвигатель, редуктор и дисковый тормоз с приводом от электромагнита постоянного тока. Электромагнит запитывается последовательно с одной из фаз обмотки электродвигателя через диоды, установленные в шкафу управления. Шкаф закреплен на корпусе механизма подъема и снабжен кнопочным подвесным пультом. По команде от пульта с удержанием кнопки электродвигатель включается, электромагнит притягивает нажимной диск тормоза, и тормоз отключается. При отпускании кнопки и выключении двигателя происходит обратное.

Виброаппаратура и аппаратно-программный комплекс (АПК) составляют единую систему (рис. 3.2), отдельные части которой сопряжены через многоканальные интерфейсы. При необходимости система позволяет вести исследования колебаний одновременно на нескольких элементах подъемно-транспортной машины. Стандартный 6-канальный виброизмерительный прибор ВИ6-6ТН (рис. 3.3, *а*), входящий в нее, работает при напряжении –27 В от настраиваемого блока питания ТЕС-18 и позволяет изменять диапазон чувствительности, производить балансировку каналов, а также выводить преобразованные сигналы по отдельным каналам на контакты выходов напряжения (U) и тока (I). К одному из каналов этого прибора подключается датчик ускорения (рис. 3.3, *б*), крепящийся на груз вдоль вертикальной оси и измеряющий вибрационные ускорения при работе механизма подъема.



в



Рис. 3.1. Таль электрическая канатная:
а – схема, главный вид; *б* – схема, вид слева; *в* – общий вид



Рис. 3.2. Виброаппаратура и АПК

Комплекс АПК состоит из двух основных совокупностей блоков:

1) типовой персональный компьютер на базе процессора Celeron 733 с подключенными ЭЛТ-монитором Falcon, матричным принтером Epson и клавиатурой, оснащенный операционной системой Windows-98 и программным обеспечением ADC8_TU (он считывает преобразованные цифровые сигналы и представляет их в графической и цифровой формах с возможностью последующей обработки);

2) нетиповые устройства сопряжения виброаппаратуры и компьютера, включающие в себя плату АЦП (рис. 3.4), устанавливаемую в слот ISA персонального компьютера (она преобразует аналоговые сигналы, поступающие на 8 каналов ее 37-контактного разъема, в цифровые) и переходную плату (рис. 3.5, а, б), содержащую набор сопротивлений и потенциометров для балансировки каналов и подключаемую, с одной стороны, экранированным кабелем к 37-контактному разъему платы АЦП, с другой – к контактам выхода виброизмерительного прибора.

а



б



Рис. 3.3. Виброаппаратура ВИБ-6ТН с датчиком виброускорения:
а – виброизмерительный прибор, *б* – датчик ускорения

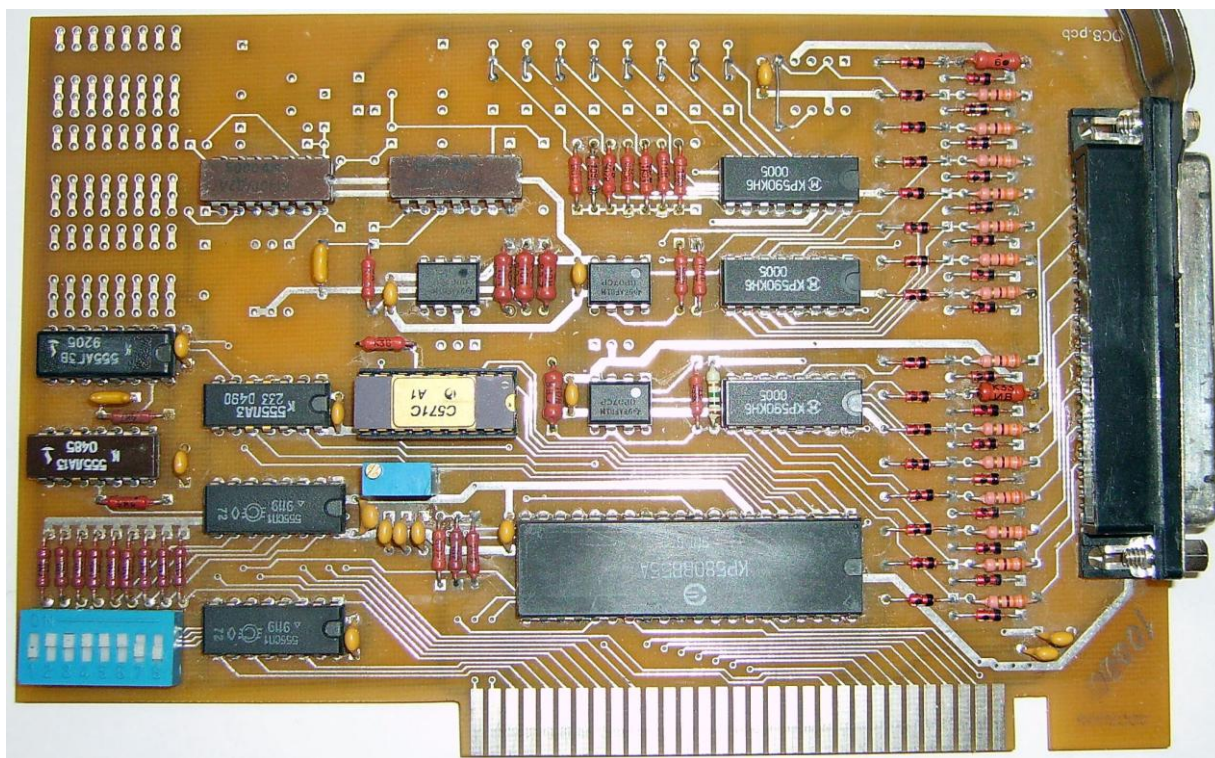


Рис. 3.4. Плата АЦП

Большой объем получаемых и обрабатываемых в процессе исследования данных и их необходимая точность предъявляют высокие требования ко всем блокам системы «виброаппаратура–комплекс АПК». И если качество и устройство типовых его частей известны, то использование нетиповых компонентов (тем более в непосредственно настраиваемой регистрирующей части) предполагает более подробное их описание. В связи с этим рассмотрим некоторые особенности применения и устройства переходной платы, платы АЦП и программной оболочки экспериментального комплекса.

Подключение датчиков или источников тока осуществляется к переходной плате или непосредственно к плате АЦП (прямое подключение не позволяет произвести подстройку эталонного напряжения). Эта плата поддерживает одновременное подключение 8 каналов, для балансировки которых на переходной плате имеются 8 потенциометров. Для каждого канала на тридцатисемиконтактных разъемах платы АЦП и переходной платы выделено по 4 контакта.

a



б

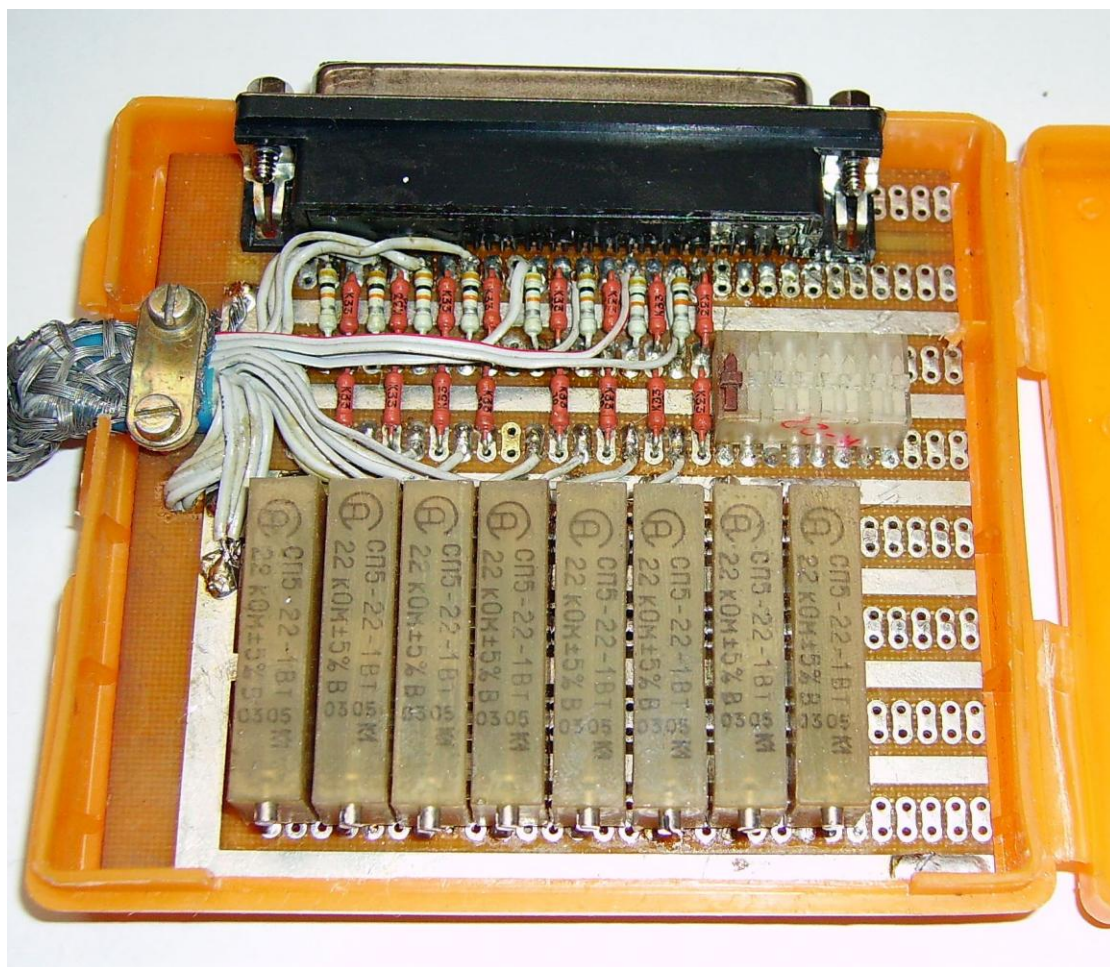


Рис. 3.5. Переходная плата для подключения к плате АЦП:
a – общий вид, *б* – конструкция

С шины компьютера через плату АЦП на переходную плату (рис. 3.6) подается постоянное напряжение 5 В. На плате АЦП установлены 2 мультиплексора КР590КН6. Оба имеют по 8 аналоговых входов (для восьми каналов) и 1 аналоговый выход, кроме того, по 3 адресные линии, устанавливаемые программно. В зависимости от опрашиваемого канала на мультиплексорах устанавливается значение адреса (от 000 до 111 в бинарном выражении). На 8 аналоговых входов одного мультиплексора подаются «эталонные» напряжения с каждого канала, а на 8 входов другого – измеряемые напряжения. Мультиплексоры коммутируют входящий сигнал с линии, адрес которой соответствует установленному, на линию аналогового выхода. Соответствующая микросхема на плате АЦП сравнивает эти два напряжения, и разница между ними записывается АЦП в цифровой форме в соответствующий регистр на плате и затем считывается программой обработки сигнала.

Для каждого канала на переходной плате имеется свой потенциометр сопротивлением 22 кОм (номера подписаны), который делит напряжение в зависимости от положения ползунка потенциометра. Крайнее положение по часовой стрелке устанавливает сопротивление верхнего плеча делителя значением 0 кОм и соответствует максимальному значению «эталонного» напряжения; крайнее положение против часовой стрелки устанавливает сопротивление верхнего плеча делителя значением 22 кОм и соответствует минимальному значению «эталонного» напряжения. Часть этого напряжения уходит на контакты 1, 3 и так далее (верхнее плечо), другая же часть – на контакты 20, 22 и так далее разъема платы АЦП (на «землю»). Между потенциометром и контактом разъема в схеме также последовательно подключено сопротивление 33 кОм (для уменьшения силы тока и снижения максимального напряжения до 4,86 В), а на плате АЦП еще по 330 Ом.

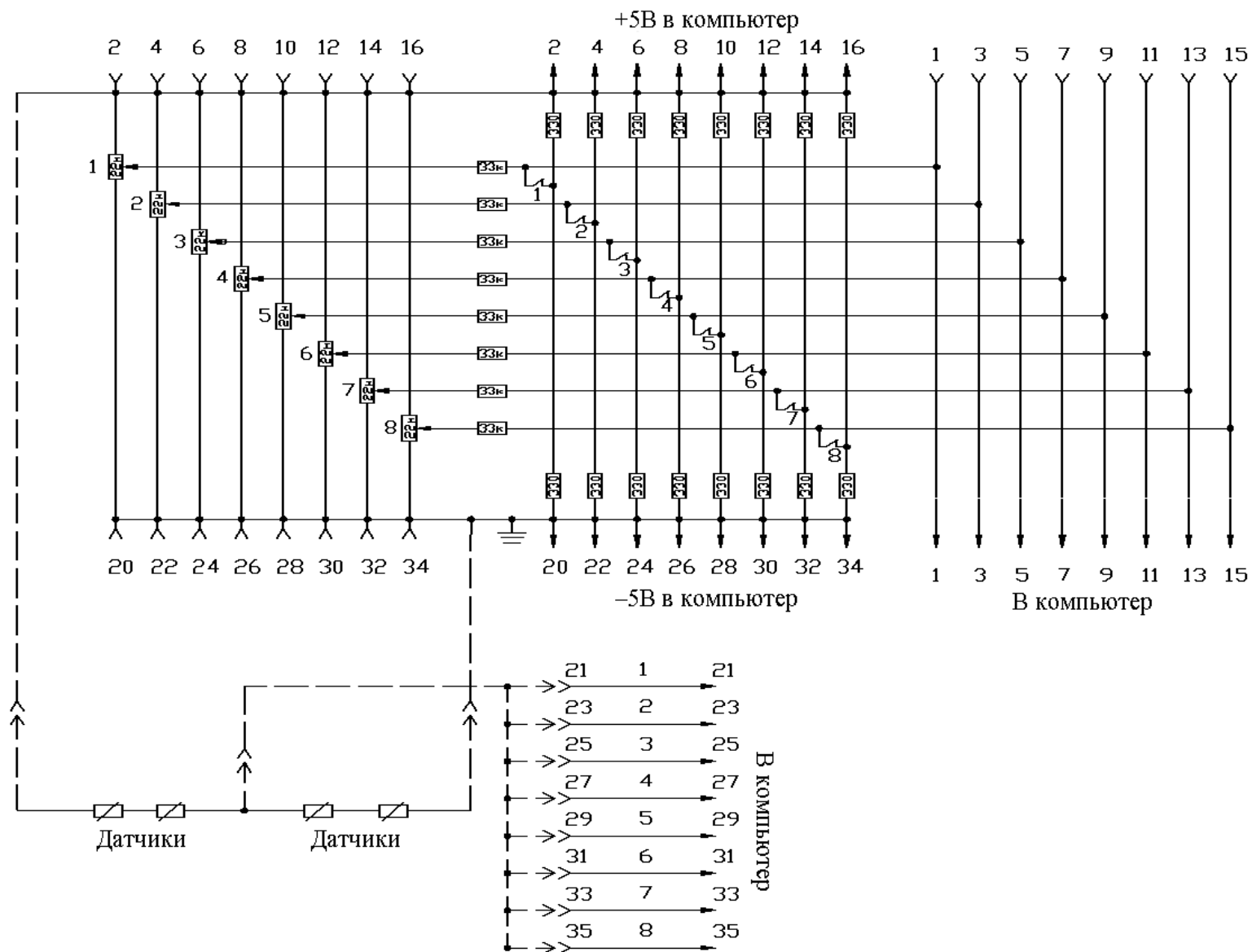


Рис. 3.6. Электрическая схема переходной платы

Параллельно плечам схемы на переходной плате возможно подключение 16 сопротивлений по 330 Ом (по 1 на плечо). Для включения сопротивления 330 Ом в схему канала на переходной плате предусмотрено 8 переключателей. Сдвинутые в сторону разъема, они замыкают среднюю точку между двумя последовательными сопротивлениями по 330 Ом с верхним плечом конкретного канала. При этом обеспечивается фактическое деление напряжения на плечах пополам вне зависимости от состояния потенциометра. При почти абсолютной идентичности начальных сопротивлений (и напряжений) на плечах измеряемой схемы или при работе только на малых коэффициентах усиления (прежде всего на первом) подобная возможность позволяет отказаться от использования ручной регулировки «эталонного» напряжения при помощи потенциометра. Однако если сопротивления в плечах подключаемой схемы изначально отличаются друг от друга, то диапазон регистрации их изменений уменьшается тем сильнее, чем выше коэффициент усиления.

Программная оболочка ADC8_TU представляет собой пользовательский интерфейс управления работой платы АЦП. Задаваемые пользователем значения (наряду со служебными) записываются программой в регистры платы перед началом работы, регистрируемые значения считываются из других регистров во время проведения эксперимента в оперативную память, а затем, по желанию пользователя, сохраняются на диск. Программа ADC8_TU была написана на языке программирования Pascal и имеется в двух версиях – оригинальной и модифицированной. В использованной при проведении эксперимента модифицированной версии введены дополнительные функции, улучшающие пользовательский интерфейс и расширяющие возможности по приведению полученных графиков к виду, в большей степени соответствующему пользовательским задачам и потребностям. Они позволяют осуществлять ряд действий, упрощающих дальнейшую обработку полученных результатов или делающих ее в принципе возможной.

После запуска программы на экране появляется ее главное окно, в котором располагается управляющее меню *Параметры работы АЦП* (рис. 3.7).

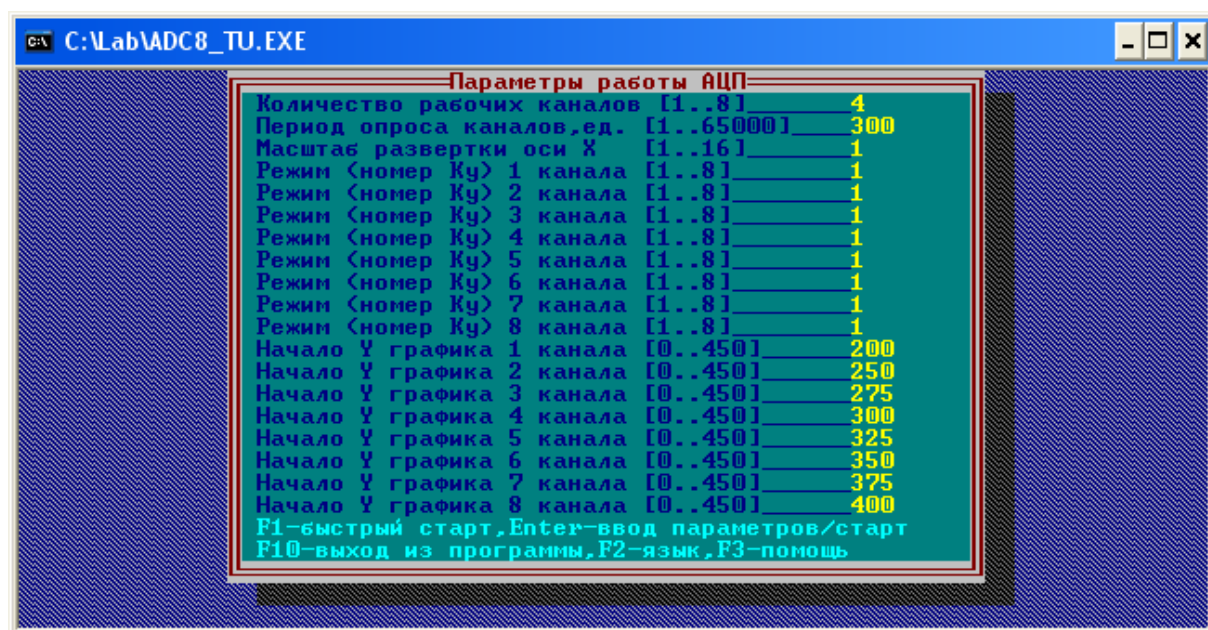


Рис. 3.7. Главное окно программы ADC8_TU

Каждый параметр имеет значение по умолчанию, указанное в соответствующем пункте при запуске программы. Допустимые значения параметров указаны в квадратных скобках.

Параметры можно разделить на пять групп:

1) *количество рабочих каналов* – вводится записываемый канал с наибольшим номером (от 1 до 8);

2) *период опроса каналов* – устанавливает задержку опроса следующего канала (порта платы АЦП) в условных единицах, реальное временное значение которых зависит от быстродействия компьютера и задается по результатам калибровки, определяющей количество секунд на одном отрезке записи сигнала (от 1 до 65 000);

3) *масштаб развертки оси X* – определяет растяжение точек опроса на экране по оси X (от 1 до 16);

4) *режим (номер КУ) канала* – устанавливает коэффициент усиления для каждого канала по отдельности и выбирается индивидуально для

конкретного случая, непосредственно завися от вероятного изменения в процессе эксперимента напряжения на плечах подключенной схемы (от 1 до 8);

5) *начало Y графика канала* – определяет первоначальное расположение графика каждого канала на экране по оси *Y* (от 0 до 450).

В главном окне для ввода параметров используются стандартные клавиши стирания значения параметра (*Backspace* (←) и *Delete*), перемещения между строками (стрелки) и фиксирования параметра с переходом на следующую строку (*Enter*), для вывода справки (*F3*), для изменения языка (*F2*), для выхода в окно записи – клавиши *F1* (из любой строки) или *Enter* (из последней строки), а для выхода из программы – *F10*.

В окне записи сигнала отображается график изменения уровня сигнала во времени и используются клавиши старта записи (*F1*), калибровки (*F2*), сохранения записи в файл (*F3*), открытия сохраненного файла (*F4*), печати (*F5*), конвертирования в Excel (*F6*), перемещения графика (*F7 – F8*), изменения масштаба (*F9 – F10*), справки (*h*) и др.

3.3. Проведение экспериментов и порядок фиксации их результатов

Весь эксперимент можно разбить на две стадии: подготовительную и собственно экспериментальную.

Подготовительная стадия включает в себя следующие этапы:

1) определение диапазонов масс груза и длин каната, при которых будут производиться измерения, шагов их изменения и количества экспериментов (на основании имеющихся данных и наличия технической возможности было принято решение проводить по одному эксперименту на опускание и подъем при диапазоне грузов от 50 до 225 кг с шагом 25 кг и

диапазоне длин каната от 100 до 200 мм с шагом, достаточным для 10 торможений на разных длинах), подготовку грузов к эксперименту;

2) подготовку таблиц, заполняемых экспериментальными данными с возможностью указания массы груза, длины каната и в дальнейшем искомой частоты колебаний при подъеме и опускании груза;

3) установку в *РС* платы АЦП с коэффициентом усиления 1 – 10000 (как более подходящей для двухканального подключения при решении поставленных задач), подключение к ее разъему переходной платы, подключение к разъему переходной платы двухканального кабеля (со стандартными трехконтактными разъемами для 3-го и 4-го каналов переходной платы и АЦП); заземление компьютера и экранированного кабеля переходной платы; подключение виброаппаратуры к блоку питания, датчика виброускорения к 4-му каналу виброизмерительного прибора, многоканального кабеля для вывода сигнала к токовому выходу I виброаппаратуры с отбором нужных контактов; заземление виброизмерительного прибора; подготовку инструмента регулировки потенциометров и балансировки каналов;

4) подключение и запуск персонального компьютера, запуск программной оболочки; подключение блока питания виброаппаратуры, установку (регулировкой ручек на блоке питания) требуемого напряжения –27 В (загорается зеленая лампочка); подключение контактов 4-го канала виброизмерительного прибора к одному из разъемов кабеля переходной платы (использовался 3-й канал АЦП и переходной платы как более надежный);

5) подключение тали к силовой сети 380 В; перемещение ее механизмом передвижения в стабильное крайнее положение над матом для постановки груза; опускание крюка, нагружение первоначальным минимальным грузом; закрепление датчика виброускорения на грузе (стрелкой на датчике вдоль вертикальной оси); заземление подвески с грузом (здесь и в предыдущих случаях заземление необходимо не только с точки зрения

техники безопасности, но, прежде всего, для устранения нежелательных помех).

К подготовительной стадии можно также отнести установку параметров, калибровку, пробный прогон, дальнейшее добавление груза и так далее, однако их можно считать и частью экспериментов.

Учитывая последнее, этапами собственно экспериментальной стадии являются следующие.

1. Установка значений всех необходимых параметров.

Необходимость установки конкретных параметров зависит от количества каналов, отличия требуемых значений от значений по умолчанию или от значений, введенных перед предыдущей записью, и т. д.

В нашем случае изменяются два параметра: *количество рабочих каналов* – 3 (поскольку вибродатчик подключаем на 3-м канале АЦП); *период опроса каналов* – 200 (чем меньше будет период опроса, тем больше точек мы получим и тем точнее можно будет определить период колебаний; однако слишком маленькое значение чревато потерей данных и некорректной работой программы). Эти значения использовались для записи всех экспериментов. При первичном поиске оптимальных значений параметров использовался и 4-й канал, и другие периоды опроса, однако указанный вариант оказался наиболее подходящим.

После установки параметров можно начинать запись сигнала. Выход в окно записи производится из любой строки меню после нажатия клавиши *F1* или из последней строки при нажатии клавиши *Enter*.

2. Калибровка АЦП.

При первоначальном выходе в окно записи сигнала (или при изменении первых трех параметров) должна быть произведена калибровка, определяющая количество секунд на одном отрезке записи сигнала (временной масштаб). Перед началом калибровки скрываем отображение графиков неиспользуемых каналов нажатием соответствующей цифровой клавиши (в нашем случае скрываем графики 1 и 2).

Калибровка запускается нажатием клавиши *F2* и останавливается нажатием любой клавиши. После остановки калибровки выводится сообщение о полученном значении количества секунд на одном отрезке записи сигнала. Здесь при нажатии клавиши *Esc* программа выходит в главное меню, а при нажатии любой другой клавиши выводится окно записи сигнала, в котором можно начинать производить запись.

Хотя при записи использовались одни и те же значения параметров, продолжительный характер эксперимента и зависимость временного масштаба от текущей загруженности операционной системы обусловили необходимость произведения калибровки перед каждым экспериментом с новым грузом, что оказалось оправданным, поскольку временной масштаб на полученных графиках при разных грузах хоть незначительно, но отличался.

3. Балансировка аппаратуры.

После проведения калибровки можно проводить регулировку потенциометров и балансировку виброизмерительного прибора. Данный этап может как следовать за окончательной калибровкой, так и предшествовать ее проведению, а также проводиться во время нее (регулировать потенциометры удобней, используя отображение изменений графика и цифровых значений для канала на экране монитора, поэтому в окне записи нажимается *F2* или *F1* и производится регулировка). Цель данного этапа – привести «эталонное» напряжение в соответствие с напряжением в экспериментальной схеме, с тем чтобы первоначальная разница между ними позволяла захватывать требуемые диапазон и направление изменений в течение эксперимента.

При входе в окно записи из главного меню в левом краю экрана, внутри расчерченной рамки записи, обозначены цифровые значения, соответствующие разнице между «эталонным» и измеряемым напряжениями для данного канала и данного КУ. Значения выведены сверху вниз (по порядку номеров каналов), отличаются цветом и располагаются по оси *Y* в

соответствии с заданными значениями начала координат. Значение 511 соответствует нулевой разнице между напряжениями. Установка в это значение при помощи потенциометров позволяет использовать диапазон от +512 до –511 последующего (во время проведения эксперимента) изменения разницы напряжений (цифровое значение для канала может меняться в пределах от 0 до 1023). При разнице напряжений, превышающей регистрируемую на данном КУ, цифровое значение приравнивается к максимальному 1023 или минимальному 0.

При осуществлении регулировки необходимое значение разницы устанавливается подкруткой потенциометра при отслеживании изменений, происходящих с графиком для данного канала на экране монитора, при этом происходящие с цифровым значением изменения отражаются в верхней правой части экрана, где они указаны теми же цветами, что и графики, и расположены в порядке следования каналов слева направо.

Подкрутка потенциометра против часовой стрелки увеличивает цифровое значение, по часовой – уменьшает. Первоначальную настройку лучше производить на 1-м КУ, поскольку, с одной стороны, на нем проще точно «поймать» значение 511, с другой – можно сразу определить разницу между текущими напряжениями и из нее установить вероятность разрыва или другой неисправности в схеме, подключенной к каналу.

Установив на 1-м КУ значение в 511, можно увеличивать КУ и уже далее производить дополнительную регулировку (если нужно).

В нашем случае для регулировки дополнительно была использована балансировка канала виброизмерительного прибора. Проверив исправность подключения, передвинув переключатель для канала в положение «к разъему» и переключившись обратно, установили подвижный контакт внутри потенциометра в среднее положение (определить которое визуально позволяет прозрачность его корпуса). Затем, предполагая, что любое отклонение от цифры 511 свидетельствует о несбалансированности виброаппаратуры, произвели ее балансировку (подогнав значение к 511). После

этого регулировку виброаппаратуры вообще не производим, используя при необходимости потенциометр переходной платы.

Однако необходимо еще провести проверку сигнала при включенной установке на используемом КУ: если значения сигнала изменяются в слишком маленьком диапазоне (визуально график не удовлетворяет требуемым задачам), то нужно КУ увеличить, и наоборот (здесь же возможно обнаружение излишних помех). Так, при включении тали с первоначальным грузом и осуществлении торможения график колебаний на экране не позволял точно определить момент завершения колебаний (да и начала тоже), поэтому режим 3-го канала был увеличен до 2-го, после чего была произведена повторная регулировка потенциометра и включение тали. На этот раз характер полученного графика был достаточен для решения требуемых задач. Дальнейшая регулировка не потребовалась – небольшие изменения уровня сигнала на протяжении всего эксперимента не играли значительной роли, будучи в рамках приемлемого.

Таким образом, при получении адекватных задаче цифровых значений уровня сигнала можно приступить непосредственно к проведению эксперимента и записи экспериментальных данных.

4. Запись сигнала.

Запись сигнала запускается нажатием клавиши *F1*. Время записи в программе ограничено размером буфера (60 000 байт), который в неизменном виде сохраняется на диск как файл. Ограничение буфера не позволяет записать в один и тот же файл данные всех экспериментов (что, впрочем, было бы и не совсем удобно). Параметры и значения записываются в буфер и файл по 2 байта. Первые 2 байта определяют количество каналов, 3-й и 4-й байты – период опроса одного канала, далее последовательно (по циклу опроса) по 2 байта записываются цифровые значения для каждого канала.

Остановка записи до окончания заполнения буфера осуществляется нажатием любой клавиши (кроме *Esc*). После этого при необходимости за-

пись может быть продолжена через определенный промежуток времени повторным нажатием *F1*. Таким образом, запись может быть произведена в один файл с остановками перед каждым последующим экспериментом.

Сохранение записи в файл после завершения экспериментов производится нажатием клавиши *F3*, после чего в появившемся окошке *Сохранить файл как ... [путь]* прописывается имя файла или полный путь и имя файла. После сохранения файла программа остается в окне записи, где можно либо продолжать работать с записью (в том числе продолжать записывать сигнал), либо, нажав клавишу *Esc*, выйти в главное меню.

При проведении экспериментов был выдержан следующий алгоритм действий:

- 1) записываем в созданную ранее таблицу массу первоначального минимального груза (50 кг);
- 2) записываем в таблицу характер перемещения груза (подъем);
- 3) перемещаем груз в соответствующее текущему характеру перемещения крайнее положение (не касаясь пола внизу или с остановом по конечному выключателю вверх);
- 4) запускаем запись сигнала в программе нажатием *F1* и соответствующее перемещение груза нажатием и удержанием кнопки на пульте управления тали;
- 5) наблюдая за поведением графика изменения виброускорения, ждем, пока затухнут колебания, вызванные началом движения;
- 6) отпускаем кнопку на пульте и ждем, пока затухнут колебания, вызванные торможением;
- 7) останавливаем запись нажатием любой клавиши (кроме *Esc*);
- 8) измеряем длину каната;
- 9) данные о длине каната записываем в таблицу;
- 10) если количество записей для данного груза и данного характера перемещения не больше девяти (т. е. десятая будет последней записью) и предполагаемое в следующей записи изменение длины каната не выйдет за

соответствующую границу крайнего положения (верхнего или нижнего), то продолжаем эксперимент в пункте 4;

11) сохраняем запись в файл нажатием клавиши *F3* с информативным уникальным именем, сокращенно содержащим характер перемещения и массу груза;

12) если последний характер перемещения – подъем, записываем в таблицу новый характер перемещения (опускание);

13) если последний характер перемещения – опускание, то, если текущая масса груза равна 225 кг, прекращаем выполнение экспериментов (сделаны все записи), если меньше 225 кг, увеличиваем массу на 25 кг и записываем в таблицу новый характер перемещения (подъем);

14) выходим в главное меню программы нажатием клавиши *Esc*, вновь заходим в окно записи нажатием клавиши *F1*;

15) производим калибровку нажатием клавиши *F2*, останавливаем и вновь выходим в окно записи двукратным нажатием любой клавиши (кроме *Esc*);

16) начинаем новую запись в пункте 3.

Таким образом, было создано 16 сохраненных на диске файлов, каждый из которых содержит информацию об изменении во времени виброускорения при торможении с данным грузом (всего 8 грузов от 50 до 225 кг) и данным перемещением (подъем или опускание).

3.4. Обработка экспериментальных данных и построение математических моделей

Данные записываются программой на диск в том виде, в котором они были представлены в памяти. При этом используется собственный формат, что не позволяет сразу начать обработку данных в других программах.

Для конвертации записей, а также их просмотра и печати в программе ADC8_TU предусмотрены соответствующие возможности.

Данные, выводимые программой на экран или на печать, берутся из предварительно заполняемого буфера ограниченного размера. Причем буфер заполняется либо в процессе записи сигнала (*F1*), либо после открытия уже сохраненного файла (*F4* и ввод имени файла).

Графическое представление и расположение данных на экране определяется использованием как отдельных параметров, вводимых в главное меню, так и различных функциональных клавиш, не только позволяющих изменить некоторые из этих параметров в окне записи, но и предоставляющих пользователю возможность выполнения различных операций над графиками.

При просмотре или перед печатью используются клавиши *F7* – *F8* и *F9* – *F10*. Клавиши *F7* и *F8* позволяют перемещаться по буферу данных на 50 значений назад и вперед (на экране это выглядит как поклеточное и одновременное перемещение всех графиков и временной шкалы влево или вправо). Таким образом можно выбрать нужный временной момент записи для его последующей печати (клавиша *F5*). Клавиши *F9* и *F10* соответствуют изменению ранее введенного параметра *Масштаб развертки оси X*. Эти клавиши позволяют двукратно уменьшать (*F9*) или увеличивать (*F10*) текущий масштаб (в последнем случае при делении нечетного числа происходит округление в меньшую сторону), что визуально соответствует сжатию или растяжению графиков и влияет в том числе на открываемые впоследствии файлы.

В модифицированной версии программы к этому добавлены функции изменения координат по оси *Y* (как одновременного для всех графиков, так и по отдельности для каждого). Для этого введен переключатель через нажатие клавиши *s* (*scrolling*), работающий по циклу: включение *Y*-прокрутки для всех графиков, включение *Y*-прокрутки по очереди для каналов 1–8 (при выключенных остальных), выключение *Y*-прокрутки для всех каналов (т. е. включение *X*-прокрутки). Прокрутка по оси *Y* осуществ-

ляется теми же клавишами, что и по оси X ($F7$ (вниз) и $F8$ (вверх)), однако здесь перемещение происходит поточечно.

Таким образом, графики уже после записи можно разместить на экране произвольным образом, что позволяет перемещать в область экрана и печати графики, вышедшие за ее пределы, расставлять графики, накладывающиеся друг на друга и т. д. При этом для отслеживания положения графиков на экране (помимо их цветового отличия) и в особенности на бумаге при печати можно использовать переключатель показа номеров графиков, реализованный через клавишу n (*numbers*). Номера располагаются слева от графиков на уровне их начала на текущем экране. Показ номеров позволяет без труда определить соответствие отдельного графика конкретному каналу АЦП, не требуя ручной отметки на бумаге.

Еще одной дополнительной функцией является возможность изменения масштаба по оси Y . Для этого используется клавиша z (*zoom*), включающая масштабирование по оси Y , при этом отключая его по оси X (и наоборот). Как и по оси X , для изменения (в 2 раза) масштаба по оси Y используются клавиши $F9$ (уменьшение) и $F10$ (увеличение), однако здесь изменения масштаба не ограничены (в том числе возможно сужение – масштаб 1:2 и менее). Текущий масштаб указывается в левом верхнем (по оси Y) и правом нижнем (по оси X) углах экрана или печатного листа.

Для равномерного распределения начал графиков на текущем экране (или просто для обеспечения быстрого появления на нем графиков, вышедших за пределы) удобно использовать (в том числе после *zoom*) клавишу a (*alignment*), автоматически выстраивающую графики по номерам сверху вниз пропорционально высоте экрана и при этом учитывающую отсутствие скрытых графиков.

Функции скрытия графиков позволяет управлять отображением как отдельных графиков, так и всех графиков вместе. Для включения/выключения отображения отдельных графиков достаточно нажать соответствующую цифровую клавишу (1–8). Для выключения всех графиков

можно использовать клавишу 0, для включения – клавишу 9. Скрытие графиков позволяет отображать только те графики на экране/бумаге, которые непосредственно исследуются.

Функция скрытия включена в том случае, когда отключена функция реверсирования. Для этого имеется соответствующий переключатель клавишей *r* (*reversing*). Реверсирование позволяет перевернуть график. Для этого используются те же цифровые клавиши, что и для скрытия.

Распределив графики на текущем экране по собственному усмотрению при помощи вышеописанных клавиш, можно приступить к печати текущей области экрана. Для этого имеется клавиша *F5*. После ее нажатия появляется окно с надписью *Печатать?* (*Print?*). Просто нажав *Enter*, пользователь запускает процесс печати текущего экрана программы. При этом возврат из процедуры печати происходит сразу же с возможностью дальнейшего проведения операций еще до того, как закончится печать.

Использования функций просмотра и печати средствами программы ADC8_TU в целом достаточно для первоначального анализа данных и даже определения периода, однако при этом имеются значительные ограничения и неудобства. В частности, отсутствует возможность выбора точки на экране и точного определения временного момента, этой точке соответствующего, что не позволяет найти точный период колебаний сигнала (приближенное же его определение «на глаз» или посредством измерения расстояний на экране линейкой будет неточным и вообще неуместным). С другой стороны, использование для этого функций печати программы тоже неприемлемо, поскольку возможности печати в ней ограничены работой с принтером, подключаемым к порту LPT и функционирующим в режиме совместимости (так, матричный принтер Epson, входящий в комплекс, позволяет распечатывать графики лишь с очень небольшой скоростью и низким качеством, приемлемыми лишь для вывода на бумагу отдельных графиков при невысоких требованиях, предъявляемых к качеству). В любом случае возможностей программы по обработке данных явно

не хватает для проведения всестороннего исследования, поэтому в ней предусмотрена конвертация формата файлов в простой формат, поддерживаемый другими программами.

Так, функция *F6* позволяет производить конвертирование файла формата ADC8_TU в текстовый, в котором цифровые значения прописываются в формате ASCII. Для значения отметки времени в программе выделено 7 знаков, в том числе три знака до запятой, запятая и три знака после запятой (соответствуют формату ##0,000, при этом свободные зарезервированные места обозначены знаком пробела). Для значения уровня сигнала на конкретном канале в момент времени, обозначенный отметкой времени, выделено пять знаков целого числа (соответствуют формату #####0).

В текстовый файл последовательно заносятся значение отметки времени (начиная с 0,000 с) и значения уровня сигнала на отдельных каналах в данный момент времени (только каналы, задействованные в процессе записи, включая графически скрытые), затем с новой строки следующий момент времени и т.д. Значения отметки времени и отдельные значения уровня сигнала разделены символами табуляции.

Данный файл может быть обработан большим количеством программ. Наиболее же удобным и простым для этого представляется пакет Excel.

Для создания конвертированного файла после нажатия клавиши *F6* в появившемся окошке *Открыть файл ... [путь]* прописывается имя открываемого файла или путь и имя файла, а в следующем окошке *Сохранить конвертированный файл как... [путь]* (*Save conversion file as... [path]*) – имя сохраняемого конвертированного файла. Для открытия файла в Excel необходимо либо задать его расширение .xls, либо открыть его непосредственно из Excel, выбрав в окне *Открытие документа* тип файла *Все файлы*.

Использование символов табуляции позволяет получить в Excel несколько столбцов данных. В столбце 1 располагаются отметки времени, во

столбцах 2–9 – значения уровня сигнала в каналах 1–8 (в зависимости от количества рабочих каналов). Пакет Excel позволяет работать с текстовыми данными как с числовыми. Наличие в нем всевозможных математических, статистических и иных функций, а также использование макросов, диаграмм и других элементов предоставляет значительные возможности по обработке записанных данных.

Одним из простейших вариантов является построение графика, подобного графику программы ADC8_TU. Для этого выделяется диапазон данных, включающих отметки времени и значения уровня сигнала, им соответствующие. Запускается *Мастер диаграмм* (через соответствующую иконку или меню *Вставка / Диаграмма...*). В появившемся окошке выбирается, например, тип *График* и *1-й вид*. После чего достаточно нажимать *Далее* и *Готово* для получения готовой диаграммы (значения в 1-й колонке выбираются как значения оси X), которая в дальнейшем может быть приведена к наиболее приемлемому варианту отображения.

Подобным образом все 16 файлов данных были конвертированы в указанный формат и сохранены с расширением *.xls*, после чего в программе Excel было получено 160 графиков изменения виброускорения во времени (по одному для каждого груза и длины каната). В дальнейшем любой из графиков можно распечатать, поместить в текст, добавить сопроводительную информацию, изменить отображение и данные, обработать встроенными функциями.

Так, поскольку виброускорение изменяется синхронно с вертикальным перемещением груза при колебаниях, частоту колебаний можно измерять непосредственно по графикам:

- 1) на графике выбирается точка, визуальна соответствующая началу колебаний при торможении;
- 2) записывается ее временное значение;
- 3) выбирается точка, соответствующая концу колебаний;
- 4) записывается ее значение;

5) подсчитывается количество колебательных пиков между двумя точками, рассчитывается количество пиков за секунду (т. е. находится искомая частота f).

Или более простым, но, возможно, менее точным способом: в середине колебательного процесса выбираются две соседние пиковые точки, разницы во времени между ними будут соответствовать периоду колебаний T , после чего по формуле $f = 1/T$ находится частота.

В результате использования первого способа значения частот были записаны в созданные ранее и набранные в Word таблицы (табл. 3.1), а в информативную часть графиков были добавлены найденные значения частот (рис. 3.8).

Если построение простых графиков, их визуальное отображение и нахождение частоты колебаний при помощи пакета Excel осуществляется весьма быстро и удобно, то выполнение более сложных задач статистической и математической обработки данных и построения моделей в нем если и реализовано, то лишь частично и менее удобно, чем в специализированных статистических и математических пакетах. В связи с этим для основной задачи – построения математической модели зависимости частоты колебаний от длины каната и массы груза – был использован пакет Statgraphics.

Statgraphics (STATistical GRAPHICs System) – статистическая графическая система, на данный момент разрабатываемая и выпускаемая фирмой StatPoint и включающая широчайший набор инструментов для статистического моделирования и анализа данных с большими графическими возможностями. При обработке данных использовалась версия программы 2006 года Statgraphics Centurion XV (до 2006 года программа называлась Statgraphics Plus и распространялась фирмой Manguistics, хотя разработчиком и тогда был StatPoint).

Таблица 3.1

Зависимость частоты f колебания груза от массы груза m ,
длины каната L и направления движения

m , кг	Опускание груза		Подъем груза		m , кг	Опускание груза		Подъем груза	
	L , м	f , Гц	L , м	f , Гц		L , м	f , Гц	L , м	f , Гц
50	1,13	12,3001	1,05	11,8009	150	1,12	9,3897	0,94	9,7847
	1,26	12,2699	1,35	12,2699		1,19	9,1158	1,04	9,5420
	1,43	11,4679	1,51	11,0742		1,34	9,2507	1,14	9,6061
	1,56	11,2532	1,59	11,9332		1,45	8,7566	1,24	9,2166
	1,72	10,9890	1,71	11,6550		1,54	8,9127	1,32	9,0498
	1,79	11,0742	1,78	11,5207		1,63	8,8731	1,41	8,9526
	1,85	11,3636	1,87	10,9170		1,76	8,5106	1,47	8,8968
	1,96	10,6496	1,95	11,6550		1,82	8,4175	1,53	8,6430
	2,09	11,4548	2,08	10,8108		1,86	8,3542	1,62	8,7413
	2,25	10,6724	2,14	10,9409		1,95	8,4818	1,79	8,4818
75	0,96	12,3001	0,88	12,6103	175	1,01	9,3809	0,91	9,4697
	1,18	12,1065	1,09	12,4844		1,15	8,9847	1,02	9,4697
	1,36	11,8624	1,23	11,7647		1,30	8,3752	1,17	9,0090
	1,40	11,6144	1,38	11,6414		1,44	8,3893	1,24	8,7873
	1,54	11,2486	1,44	11,2233		1,50	8,1037	1,34	9,0827
	1,63	10,8932	1,57	11,0742		1,60	8,4104	1,44	8,1900
	1,80	10,4275	1,68	11,1857		1,72	8,1169	1,50	8,3542
	2,06	10,1420	1,76	11,1483		1,78	8,1699	1,61	8,2850
	2,14	9,8522	1,78	10,7296		1,87	8,4317	1,70	8,0645
	2,30	9,8039	1,96	10,8108		1,90	7,8802	1,74	8,1301
100	1,08	11,4155	0,92	12,3762	200	1,01	8,9127	0,86	9,4697
	1,22	11,1982	1,04	11,5473		1,10	9,2507	1,09	8,8106
	1,32	10,5263	1,19	11,1607		1,29	8,4246	1,16	8,5837
	1,40	10,4822	1,35	10,2041		1,32	8,4962	1,26	8,8889
	1,48	9,6712	1,44	10,0301		1,41	8,1235	1,35	8,3542
	1,60	10,2459	1,54	10,2564		1,47	8,2372	1,44	8,1699
	1,67	9,8912	1,60	10,1626		1,52	8,4175	1,51	8,1169
	1,74	9,7182	1,70	9,6712		1,62	7,9681	1,61	8,3195
	1,92	9,4251	1,79	9,6618		1,70	7,7760	1,69	7,8802
	2,03	9,6154	1,86	9,6339		1,90	7,4963	1,79	7,9051
125	0,90	11,2486	1,00	10,5708	225	1,00	8,6730	0,92	8,9526
	1,07	11,0497	1,08	10,4275		1,08	8,9767	1,10	8,6059
	1,18	10,5820	1,22	10,1420		1,18	8,8028	1,21	8,2645
	1,31	10,2564	1,32	9,8912		1,27	8,6806	1,28	8,2372
	1,39	9,8039	1,42	9,8717		1,35	8,0580	1,32	8,3333
	1,60	9,4697	1,48	9,6805		1,47	7,7340	1,40	8,0064
	1,67	9,4251	1,62	9,5420		1,55	7,8864	1,48	8,0972
	1,75	9,1827	1,75	9,3371		1,65	7,6104	1,58	7,9302
	1,90	9,0009	1,80	9,1743		1,77	7,6805	1,60	7,9114
	2,02	8,9606	1,88	9,0416		1,88	7,0922	1,69	7,5758

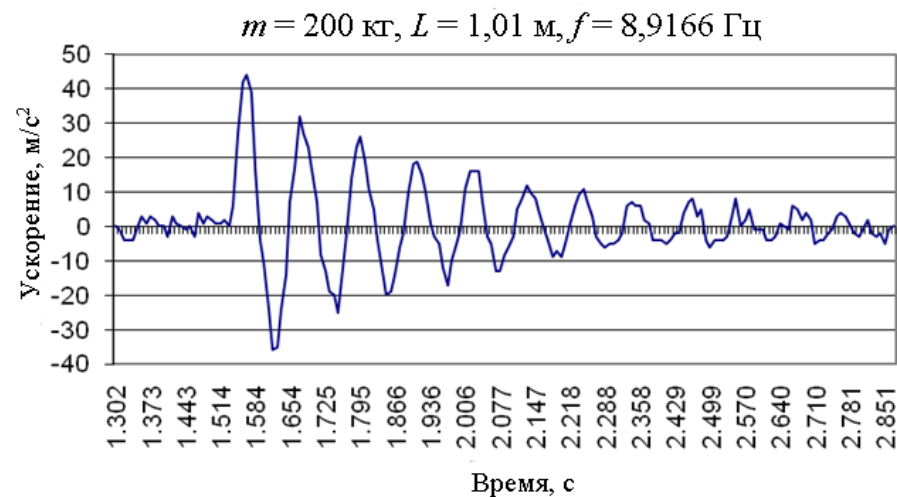
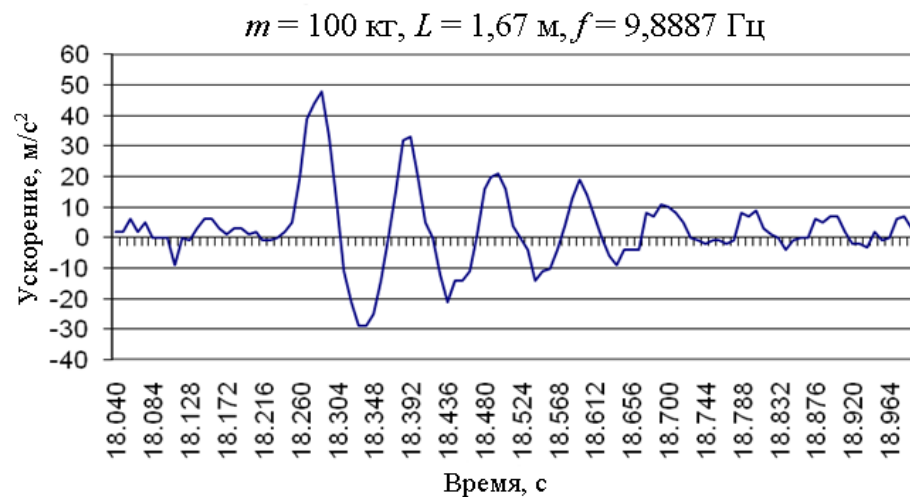
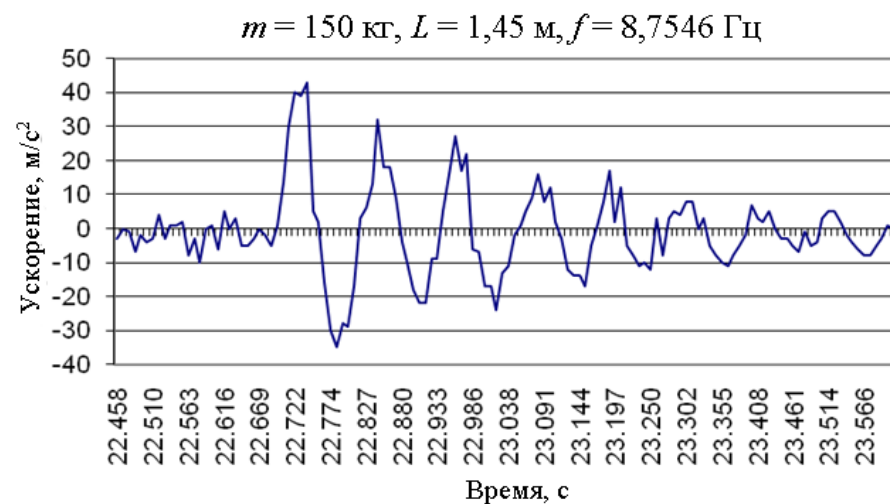
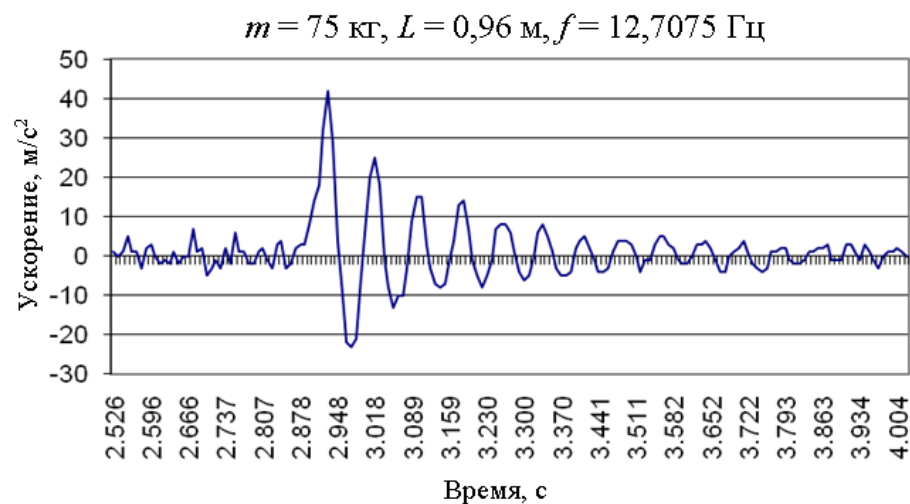


Рис.3.8. Образцы экспериментальных графиков зависимости виброускорения и частоты колебания от длины каната и массы груза

Продукт Statgraphics присутствовал на рынке, начиная с 1980 года: вначале для операционной системы DOS, с 1994 года – для системы Windows. Statgraphics Centurion XV включает более 150 статистических процедур, применяющихся в бизнесе, экономике, маркетинге, машиностроении, медицине, биологии, социологии, психологии и других областях.

Основными преимуществами пакета являются:

- интуитивный интерфейс и динамическая обработка данных, позволяющие значительно сократить время, затраченное на анализ;
- автоматическая интерпретация результатов обработки данных в точных, понятных терминах;
- возможность обработки статистических данных различного качества и из различных источников;
- всевозможные графики, представляющие интегрированную часть анализа и расширяющие его возможности в тех случаях, когда табличные данные недостаточно информативны или слишком сложны;
- всесторонние результаты анализов, сохраняемые в различных форматах, любом объеме и любой форме;
- возможность применения при обработке и интерпретации данных системного подхода (Шести сигм).

Уникальность пакету придают и его инструменты:

StatWizard – помогает выбрать наиболее подходящие процедуры статистического анализа для обрабатываемого типа данных;

StatAdvisor – формирует мгновенные, доступные для понимания интерпретации статистических результатов;

StatReporter – объединяет результаты анализа и создает качественный отчет-презентацию;

StatGallery – позволяет объединять на одной странице до 9 графиков, а также накладывать 2 графика друг на друга для визуального сравнения;

StatFolios – сохраняет анализы с возможностью их дальнейшего изменения или отражения изменений при появлении новых данных;

StatPublish – публикует данные в сети Интернет;

StatLink – связывает анализ с источником данных для автоматического отражения их текущих изменений;

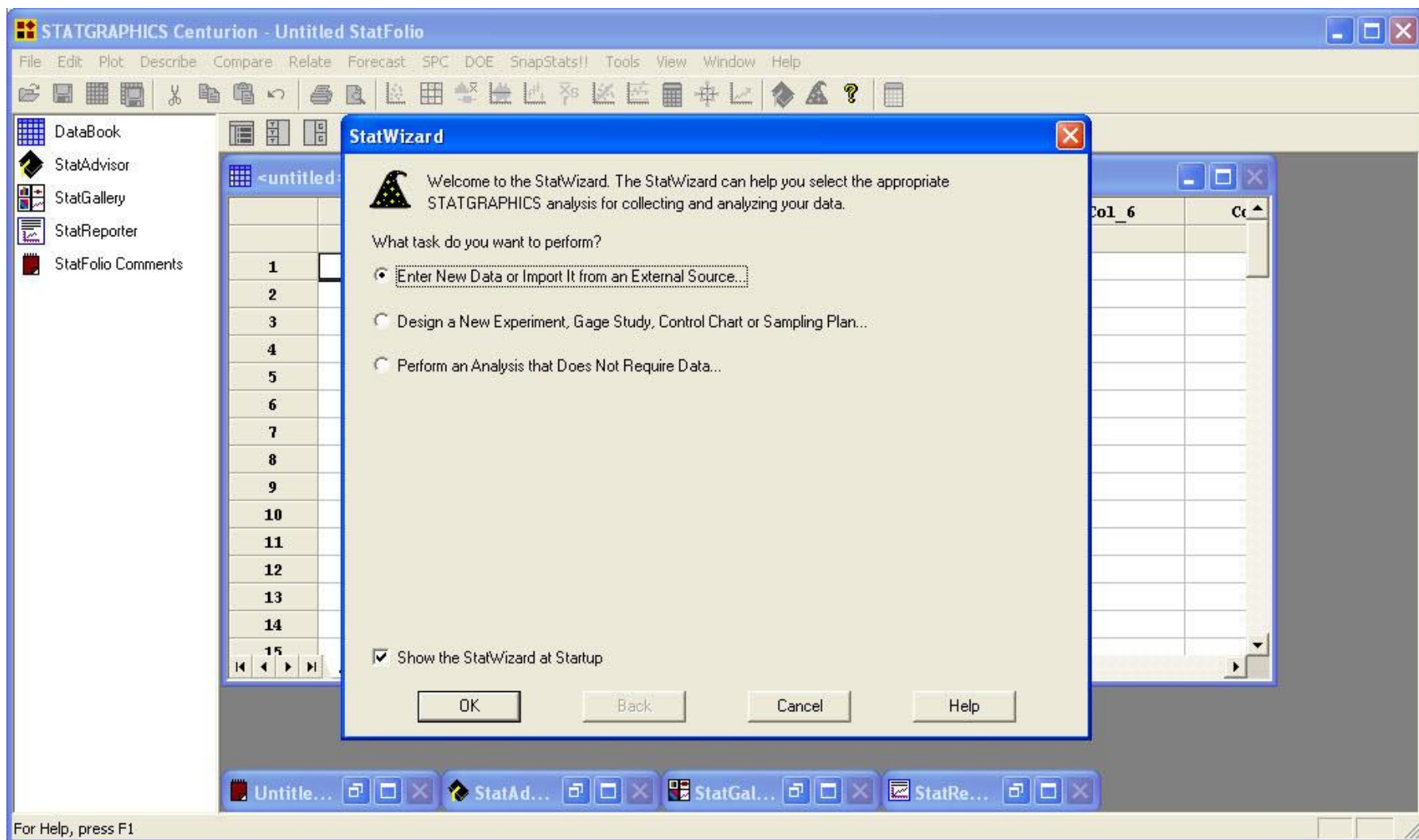
SnapStats – формирует компактные версии наиболее используемых процедур анализа с целью помещения результатов на одну страницу.

При запуске программы появляется помощник *StatWizard* (рис. 3.9), пошагово создающий новый проект. При этом на первом шаге можно выбрать в качестве текущей задачи либо проведение анализа без ввода данных, либо разработку нового мониторингового исследования, либо ввод данных для последующей обработки (как раз то, что нас интересует).

В последнем случае на следующем шаге пользователю предлагается выбрать источник данных, которые будут располагаться в ячейках стандартного табличного интерфейса программы. Statgraphics поддерживает большое количество форматов данных и способов их попадания в ячейки. Данные можно вводить с клавиатуры вручную, вставлять из буфера обмена Windows, подключать через соединение с базой данных, использовать внешние (файлы стандартных форматов) и внутренние (собственный формат) источники данных. При любом подходе (в том числе при отмене использования *StatWizard*) пользователь в конечном счете попадает в книгу данных (*DataBook*) программы. Непосредственно там данные можно начинать вводить либо обрабатывать.

Так, для нашего исследования в ячейки программы были введены данные времени (столбец 1) и виброускорения (столбец 2), записанные во время эксперимента при помощи ADC8_TU (рис. 3.10). Для этого конвертированный файл формата Excel либо загружается напрямую в программу (без права изменения данных во внешнем источнике), либо данные копируются через буфер обмена.

Как видно из рисунка, для доступа к своим процедурам Statgraphics предоставляет целую серию меню и пиктограмм, расположенных в верх-

Рис. 3.9. Помощник *StatWizard* при старте программы

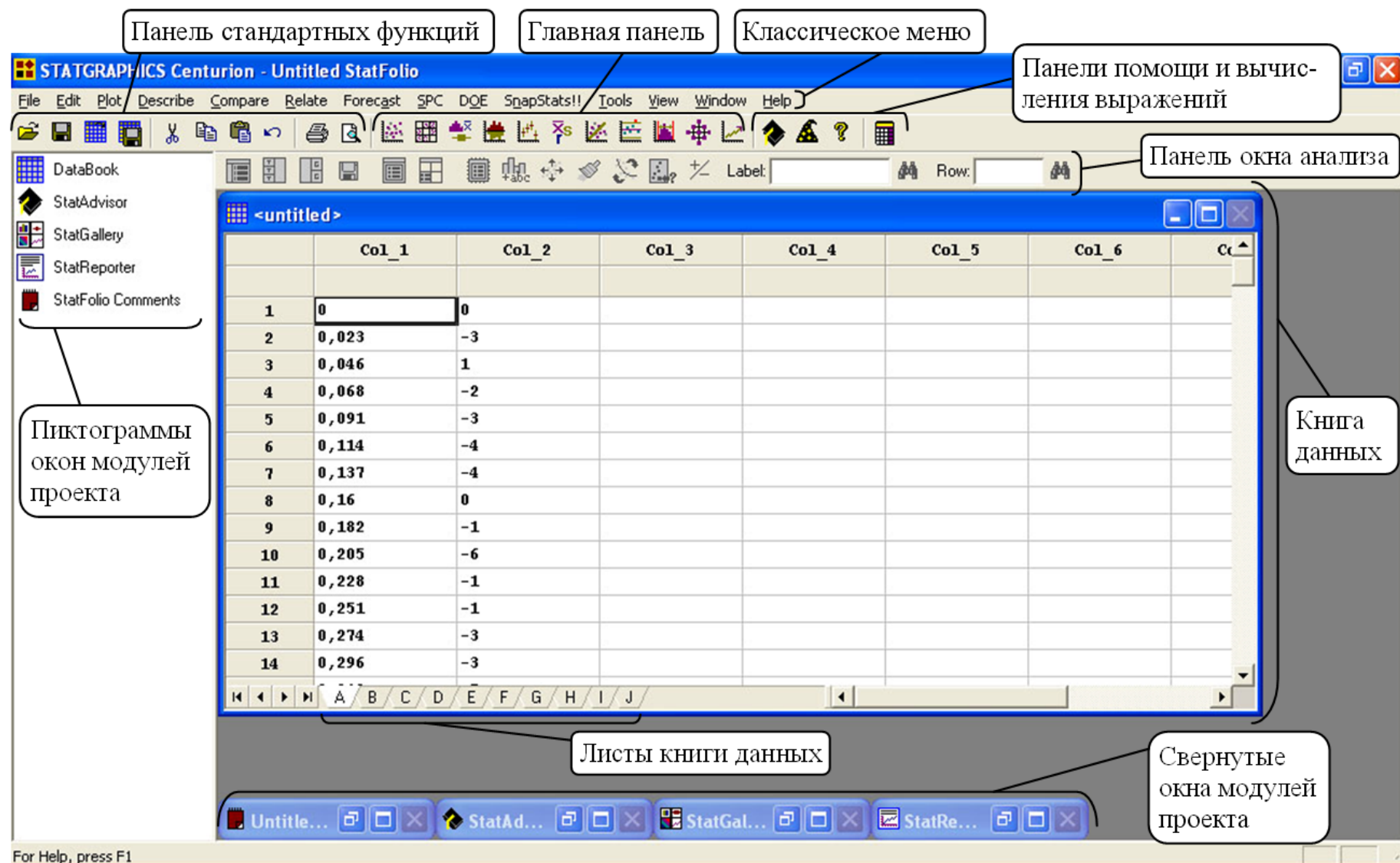


Рис. 3.10. Окно программы Statgraphics с открытым окном книги данных

ней части экрана. Причем существует целых два набора меню: классический и Шести сигм. И тот и другой охватывает все возможности, но в соответствии с их концепцией процедуры распределяются и называются по-разному. Так, идеология Шести сигм представляет собой высокотехнологичную методику точной настройки бизнес-процессов, применяемую с целью минимизации вероятности возникновения дефектов в операционной деятельности, и строится на парадигме DMAIC – *Define, Measure, Analyze, Improve, Control* (или в русском варианте ОИАСК – Определение, Измерение, Анализ, Совершенствование, Контроль). В программе в виде меню присутствуют все 5 ступеней DMAIC. Однако поскольку использование программы Statgraphics для целей данной работы ограничивается процедурами анализа и, возможно, измерения, то и применяться будет более удобный для несистемных применений пакета классический вариант меню. Переключение между вариантами осуществляется через окно программных настроек (*Edit / Preferences...*), которое позволяет изменить множество значений по умолчанию, относящихся как к программе в целом, так и к ее отдельным процедурам.

Помимо стандартного набора функций работы с файлами (*File*), редактирования (*Edit*), изменения вида окна (*View*), переключения окон (*Window*) и справки (*Help*), классическое меню содержит процедуры построения графиков (*Plot*), описания данных (*Describe*), сравнения данных (*Compare*), отношения данных (*Relate*), прогнозирования (*Forecast*), статистического контроля процессов (*SPC – Statistical Process Control*), дизайна экспериментов (*DOE – Design Of Experiments*), а также вспомогательные меню компактного снимка аналитических процедур (*SnapShots!!*) и инструментов (*Tools*).

Меню *Plot* предоставляет возможность построить следующие виды графиков: диаграммы рассеивания (*Scatterplots*), разведочные графики (*Exploratory Plots*), временные графики (*Time Sequence Plots*), деловые карты (*Business Charts*), диаграммы распределения вероятностей (*Probability Dis-*

tribution), сплины (*Splines*), графики поверхностей и контуров (*Surface and Contour Plots*), пользовательские диаграммы (*Custom Charts*). Оно предназначено только для создания графических отображений данных без использования аналитических процедур.

Меню *Describe* содержит статистические методы общего анализа по одной или множеству переменных и позволяет анализировать числовые (*Numeric Data*) и категориальные (*Categorical Data*) данные, тестировать жизненные циклы (*Life Data*), подбирать распределения (*Distribution Fitting*), находить корреляцию (*Multivariate Methods*), описывать временные ряды (*Time Series*).

Меню *Compare* включает в себя методы сравнения двух (*Two Samples*) и более (*Multiple Samples*) выборок данных, процедуры дисперсионного анализа (*Analysis of Variance*).

Меню *Relate* позволяет выявить соотношение между переменными в одно- (*One Factor*) и многофакторных (*Multiple Factors*) моделях, проанализировать данные о параметрах процесса (*Attribute Data*) и его жизненного цикла (*Life Data*), произвести классификацию данных (*Classification Methods*).

С помощью меню *Forecast* возможно спрогнозировать дальнейшее поведение рассматриваемых величин на основании имеющихся данных и в соответствии с моделью, либо заданной пользователем (*User-Specified Model*), либо определенной автоматически (*Automatic Model Selection*).

Процедуры меню статистического контроля процессов (*SPC*) позволяют анализировать качество выходного продукта (*Quality Assessment*) и производительность процесса (*Capability Analysis*), выстраивать диаграммы контроля (*Control Charts*), разрабатывать методику измерений и анализировать данные мониторинга (*Gage Studies*), осуществлять выборочный приемочный контроль (*Acceptance Sampling*).

Меню *DOE*, посвященное дизайну эксперимента, делает возможным как создание этого дизайна (*Design Creation*), так и проведение его качест-

венного анализа (*Design Analysis*).

Процедуры меню дублируются 11 пиктограммами главной панели инструментов (*Toolbar*), расположенной в середине экрана. По умолчанию используется следующий набор наиболее востребованных, с точки зрения разработчика, элементов: диаграмма рассеивания $X - Y$ (*X - Y Scatterplot*), матричная диаграмма (*Matrix Plot*), анализ одной переменной (*One Variable Analysis*), сравнение двух выборок (*Two Sample Comparison*), сравнение нескольких выборок (*Multiple Sample Comparison*), анализ нескольких переменных (*Multiple Variable Analysis*), простая регрессия (*Simple Regression*), диаграмма индивидуальных измерений и скользящего диапазона (*X and MR(2) Chart*), анализ производительности по отдельным элементам (*Capability Analysis (Individuals)*), создание дизайна (*Create Design*), автоматический прогноз (*Automatic Forecasting*). Любой из 11 пиктограмм может быть назначено иное значение из широкого выбора имеющихся процедур посредством выбора меню *Edit / Toolbar Shortcuts...*

Помимо главной панели, присутствуют панели со стандартными функциями открытия–сохранения файлов, редактирования, печати. Кроме того, имеется дополнительная панель помощи с возможностью выбора между *StatAdvisor*, *StatWizard* и стандартным *Help*, а также панель с пиктограммой *Evaluator* – своеобразного калькулятора Statgraphics, позволяющего вычислить выражение с использованием выбранных переменных и встроенных операторов и функций.

Под набором панелей основного ряда над рабочим окном программы расположена панель, пиктограммы которой позволяют, с одной стороны, изменить (в рамках текущего анализа) входные данные, методы их обработки и средства отображения результатов, с другой – интерактивно исследовать и изучать полученный анализ. Эти функции доступны только при активации окна анализа, которое формируется при запуске любой из процедур Statgraphics и разделяется на несколько частей, в каждой из ко-

торых помещаются заданные по умолчанию таблицы (слева) и графики (справа).

Так, функция ввода данных (*Input Dialog*) изменяет задействованные в анализе переменные либо обрабатываемый диапазон данных. Пиктограммы *Tables* и *Graphs* позволяют выбрать для текущей процедуры дополнительные таблицы и графики. Пиктограмма *Save results* сохраняет полученные результаты анализа в отдельных именованных столбцах любых листов рабочей книги. Функция же *Analysis Options* изменяет параметры текущего анализа, что непосредственно отражается в полученных результатах как табличного, так и графического вида.

При двойном щелчке мышью по интересующей таблице или графику выбранная часть расширяется, занимая все окно анализа, что позволяет не только рассмотреть результаты в большем масштабе, но и получить доступ к дополнительным элементам описываемой панели (только для графической части).

Так, функции *Pane Options* и *Graphics Options* изменяют параметры графического представления результатов, причем первая индивидуальна по составу и возможностям для каждого из графиков, а вторая содержит единый для всех набор атрибутов, таких как используемый профиль, палитра, легенда, название, макет, сетка, линии, а также диапазон и шаг отображаемых на графике данных. С той же функцией визуального оформления графика выступает пиктограмма *Add text* (добавление текста с последующей возможностью его перемещения и изменения). Процедура *Jitter* (*Случайные искажения*) изменяет (случайным образом и незначительно) расположение точек с целью предотвратить их наложение друг на друга при равных значениях. Процедура *Brush* (*Кисть*) окрашивает точки, попадающие в выбранный диапазон значений, а функция *Smooth/Rotate* сглаживает 2-мерный график или вращает 3-мерный.

Для графика в полноэкранном режиме при щелчке правой кнопки мыши становятся доступными дополнительные возможности (помимо

стандартных функций копирования, печати, сохранения и настроек). Они позволяют выделить объект (с целью перемещения и изменения размеров) или точку данных (*Select*; по умолчанию), установить расположение точки графика с точностью до 0,00001 (*Locate*), рамкой увеличить (*Zoom In*) или уменьшить (*Undo Zoom*) интересующую область графика.

Выделение точки данных делает возможным удаление ее с графика при помощи пиктограммы той же панели +/- (либо возвращение точки назад), а также определение ее значения, номера (*Row*) и названия (*Label*) ряда, в котором она расположена, при этом под названием подразумевается любое (в том числе и числовое) значение, расположенное в том же ряду. Столбец названий задается пиктограммой *Identify*, после чего каждая точка на графике получает собственное идентификационное значение, предоставляя пользователю возможность сразу получить дополнительную информацию по ней, не прибегая к просмотру указанного ряда.

Помимо окон анализа и книги данных, в Statgraphics встроены оконные интерфейсы *StatAdvisor*, *StatGallery*, *StatReporter* и *StatFolio Comments*. В окне *StatAdvisor* отображаются аналитические выкладки, созданные этим инструментом и сопровождающие каждую таблицу. *StatGallery* позволяет сгруппировать до 9 графиков из окна анализа по различным макетам их расположения. *StatReporter* принимает как графики, так и таблицы с аналитикой и создает отчет презентационного характера. В окне же *StatFolio Comments* помещаются комментарии общего характера, описывающие текущий проект на всех стадиях его реализации.

Все эти окна во время работы и в неактивном состоянии представлены как в сложенном виде внизу экрана, так и в виде удобных ссылок слева, что является немаловажным, учитывая необходимость использования, зачастую, многочисленных процедур и важность быстрого доступа к отдельным окнам при проведении всестороннего анализа. Окно *StatGallery* требует своего сохранения в файле собственного типа с разрешением *.sgg, окно *StatReporter* – в стандартном текстово-графическом файле формата

*.rtf, книга данных при ее ручном вводе или копировании – в собственном формате *.sf6, окна анализа сохраняются в виде процедур внутри файла проекта *StatFolio* (наряду с комментариями), создаваясь заново при открытии проекта и отражая возможные изменения источника данных или самих данных (в *StatFolio* сохраняется лишь ссылка на них).

Если данные были скопированы или впечатаны в рабочую книгу, то первым делом нужно дать значимые имена ее элементам, а также сохранить проект и данные во избежание случайных потерь (некоторые из этих действий могут быть выполнены на одном из этапов работы *StatWizard* при его использовании). Так, пробные введенные данные по эксперименту были сохранены в файл *data.sf6*, аналогично названию проекта *StatFolio* – *data.sgp*. Затем, дважды щелкнув по первому, а затем второму столбцу, установили в открывшемся окне их имена (здесь же тип данных задается как числовой (*Numeric*)), а поле комментариев оставили пустым). Таким образом, столбец данных со временем получил название *Time*, столбец с ускорением – *Acceleration*, а дальнейшая задача – выбрать из множества процедур подходящие для нашего случая.

Для определения точек начала и окончания колебаний достаточно использовать построенный по ним график. Поскольку значения точек изменяются во времени, то в меню *Plot* выбирается подменю графиков временной последовательности (*Time Sequence Plots*), где искомым для нас будет обычный график одной переменной (*X-plot*).

Для анализа временного ряда с целью определения частоты колебаний используются описательные (дескриптивные) методы из меню *Describe / Time Series / Descriptive Methods...* В частности, метод этого анализа (периодограмма) позволяет определить частоту почти автоматически.

Однако подобное нахождение частоты все-таки не превосходит по скорости и удобству (да и по качеству) вычисление ее с использованием графического отображения в том же Excel. Где действительно проявляются преимущества Statgraphics, так это в приложении ее богатых аналитиче-

ских возможностей, например, к отысканию зависимостей между переменными и построению для них оптимальной математической модели – нетривиальной задачи, решение которой сложно произвести не только математическими, но и программными средствами. Для этого в Statgraphics широко используются процедуры меню *Relate* и в особенности те из них, которые находят одно- и многофакторные модели посредством регрессионного анализа.

Для построения математической модели зависимости частоты колебаний от длины каната и массы груза в новую рабочую книгу Statgraphics были скопированы данные из табл. 3.1 (причем вначале – данные для опускания, затем ниже – для подъема). Точнее сказать, только данные длины каната (второй столбец книги) и частоты колебаний (третий столбец). Каждое общее для 10 экспериментов (и строк) значение массы нужно было поместить в 10 отдельных строк книги данных. Для этого можно было либо поклеточно вставлять значения, либо использовать встроенную процедуру Statgraphics – *Generate Data* (Генерировать данные). Данные генерируются для полностью выделенного столбца, соответствующее же диалоговое окно вызывается из меню *Edit*, после чего в поле *Expression* (Выражение) вводится нужная формула. В нашем случае она выглядела следующим образом: *RESHAPE(REP(COUNT(50;225;25);10);160)*. В ней встроенный оператор *RESHAPE* формирует массив данных размером 160 единиц (2-й аргумент оператора) и заполняет его генерируемым шаблоном (1-й аргумент), причем при незаполнении массива операция циклично повторяется до окончательного заполнения. В качестве генератора значений шаблона выступает оператор *COUNT*, на выходе которого выдаются числа от 50 до 225 с шагом 25 (аргументы оператора 1, 2, 3), каждое из которых последовательно повторяется оператором *REP* (repeat) 10 раз (второй аргумент оператора). Эти два оператора выдают 80 чисел за первый цикл, после чего еще раз 80 чисел для заполнения массива в 160 единиц.

После добавления данных 1-й столбец масс был назван m , 2-й столбец длин – L , 3-й столбец частот – f . Подобные сокращенные названия удобны, поскольку, с одной стороны, являются известными обозначениями рассматриваемых величин, с другой – позволяют получать формулы модели в уже готовой математической форме. Кроме того, в поле *Comments* каждого столбца были добавлены расшифровки обозначений: *mass*, *Length* и *frequency*. Книга данных была сохранена под именем *model.st6*, а проект *StatFolio* как *model.sgp*. Рабочий лист был также обозначен как *model*, после чего получили готовую к обработке книгу данных (рис. 3.11).

В качестве простейшего (с точки зрения удобства) инструмента регрессионного анализа в пакете имеется процедура простой регрессии (*Simple Regression*), доступ к которой осуществляется через меню *Relate / One factor* или через пиктограмму главной панели. Как следует из названия меню, процедура простой регрессии заключается в нахождении аналитического выражения для связи двух переменных – независимой (или фактора) X и зависимой Y . В результате вызова процедуры в окне анализа публикуется аналитическая сводка, включающая готовую модель, показатель корреляции с экспериментальными данными *R-squared* или *R-квадрат* и при необходимости упорядоченное по значению *R-squared* сравнение сразу двадцати семи моделей. Однако поскольку речь идет только об одном факторе, а в нашем случае их два, то для поставленной задачи простая регрессия неприменима.

В связи с этим необходимо использовать операции множественной регрессии (меню *Relate / Multiple Factors*).

Предметом множественного регрессионного анализа является установление статистической зависимости величины Y от нескольких других величин X_1, X_2, \dots, X_n . Из всех операций вышеупомянутого меню наибольший интерес представляют две: *Multiple Regression* (Множественная регрессия) и *Nonlinear Regression* (Нелинейная регрессия). С первой и начиналась аналитическая обработка наших данных.

STATGRAPHICS Centurion - model.sgp - [model.sf6]

File Edit Plot Describe Compare Relate Forecast SPC DOE SnapStats!! Tools View Window Help

DataBook StatAdvisor StatGallery StatReporter StatFolio Comments

Label: Row:

	m	L	f	Col_4	Col_5	Col_6	Col_7	Col_8
	mass	Length	frequency					
1	50	1,13	12,3001					
2	50	1,26	12,2699					
3	50	1,43	11,4679					
4	50	1,56	11,2532					
5	50	1,72	10,9890					
6	50	1,79	11,0742					
7	50	1,85	11,3636					
8	50	1,96	10,6496					
9	50	2,09	11,4548					
10	50	2,25	10,6724					
11	75	0,96	12,3001					
12	75	1,18	12,1065					
13	75	1,36	11,8624					
14	75	1,4	11,6144					
15	75	1,54	11,2486					
16	75	1,63	10,8932					
17	75	1,8	10,4275					
18	75	2,06	10,1420					
19	75	2,14	9,8522					
20	75	2,3	9,8039					
21	100	1,08	11,4155					

model B C D E F G H I J

For Help, press F1

Рис. 3.11. Книга данных модели зависимости частоты колебаний от массы груза и длины каната

При запуске этой процедуры предлагается выбрать Зависимую переменную (*Dependent Variable*) и Независимые переменные (*Independent Variables*). Кроме того, здесь же в поле *Select* устанавливается диапазон рядов для обработки. Поскольку математическая модель находилась для трех вариантов (при опускании, при подъеме и при опускании и подъеме), то для каждого варианта это поле было заполнено по-разному: для опускания введено *FIRST(80)* – первые 80 данных, соответствующих опусканию, для подъема установлено *LAST(80)* – последние 80 при подъеме, а в третьем случае поле осталось пустым (т. е. использовались данные всего диапазона).

В результате получили окно анализа, разбитое на 4 части (рис. 3.12): в табличной части – Сводка анализа (*Analysis Summary*) и Необычные остатки (*Unusual Residuals*), в графической – графики отношения Наблюдений и предсказаний (*Observed versus Predicted*) и Остатков и предсказаний (*Residuals versus Predicted*). Как и раньше, можно выбрать другие виды таблиц и графиков. Кроме того, в окне *Analysis Options* имеется возможность применить к модели метод пошаговой регрессии, целью которого является, постепенно добавляя переменные в модель (*Forward Selection*) либо удаляя их из модели (*Backward Selection*), достичь оптимального состояния, оставив в результате в модели лишь те переменные, которые статистически значимы.

При всем ее удобстве данная процедура посвящена исключительно линейной регрессии и, в отличие от процедуры простой регрессии, описанной выше, не позволяет провести исследования альтернативных моделей (очевидно, по причине ресурсоемкости либо неоднозначности подобного функционала для множественной регрессии), ограничиваясь линейной. Значение коэффициента *R-квадрат* для нее, хотя и превышает 90 %,

но до 95 % не дотягивает, определяя необходимость нахождения более качественных моделей среди нелинейных.

Для этого использовалась процедура *Nonlinear Regression*, которая в конечном итоге и выдала искомые результаты (на самом деле, название «нелинейная» в данном случае описывает только возможность использования нелинейных функций, не исключая при этом и линейные).

При запуске процедуры выдается диалоговое окно, в котором в поле *Dependent Variable* вносим переменную f , а в поле *Function* – аппроксимирующую функцию (модель), при обработке текста которой буквенные значения, совпадающие с названиями столбцов, интерпретируются как переменные, а не совпадающие – как числовые константы, поиском оптимальных значений которых процедура и занимается.

Таким образом, предварительно были отобраны 6 функций вида $f = F(m, L)$: прямая линейная, экспоненциальная, степенная, две обратные линейные и полиномиальная. Символьное представление каждой помещалось в поле *Function*, после чего формировалось окно анализа.

В качестве примера рассмотрим выполнение и результаты процедуры регрессионного анализа зависимости вертикальных колебаний, возникающих после торможения при подъеме груза, от длины каната и массы груза с нахождением математической модели этой зависимости при описании ее полиномиальной функцией.

После запуска процедуры в поле *Select* окна ввода данных занесли зависимую переменную частоты колебаний f . В поле *Function* набрали в символьном виде правую часть полиномиальной функции

$$a + bm + cL + dm^2 + eL^2 + gmL,$$

где m – масса груза; L – длина каната; a, b, c, d, e, g – искомые константы модели. В поле же *Select* занесли формулу *LAST(80)* – последние 80 рядов, соответствующие процессу подъема (рис. 3.13).

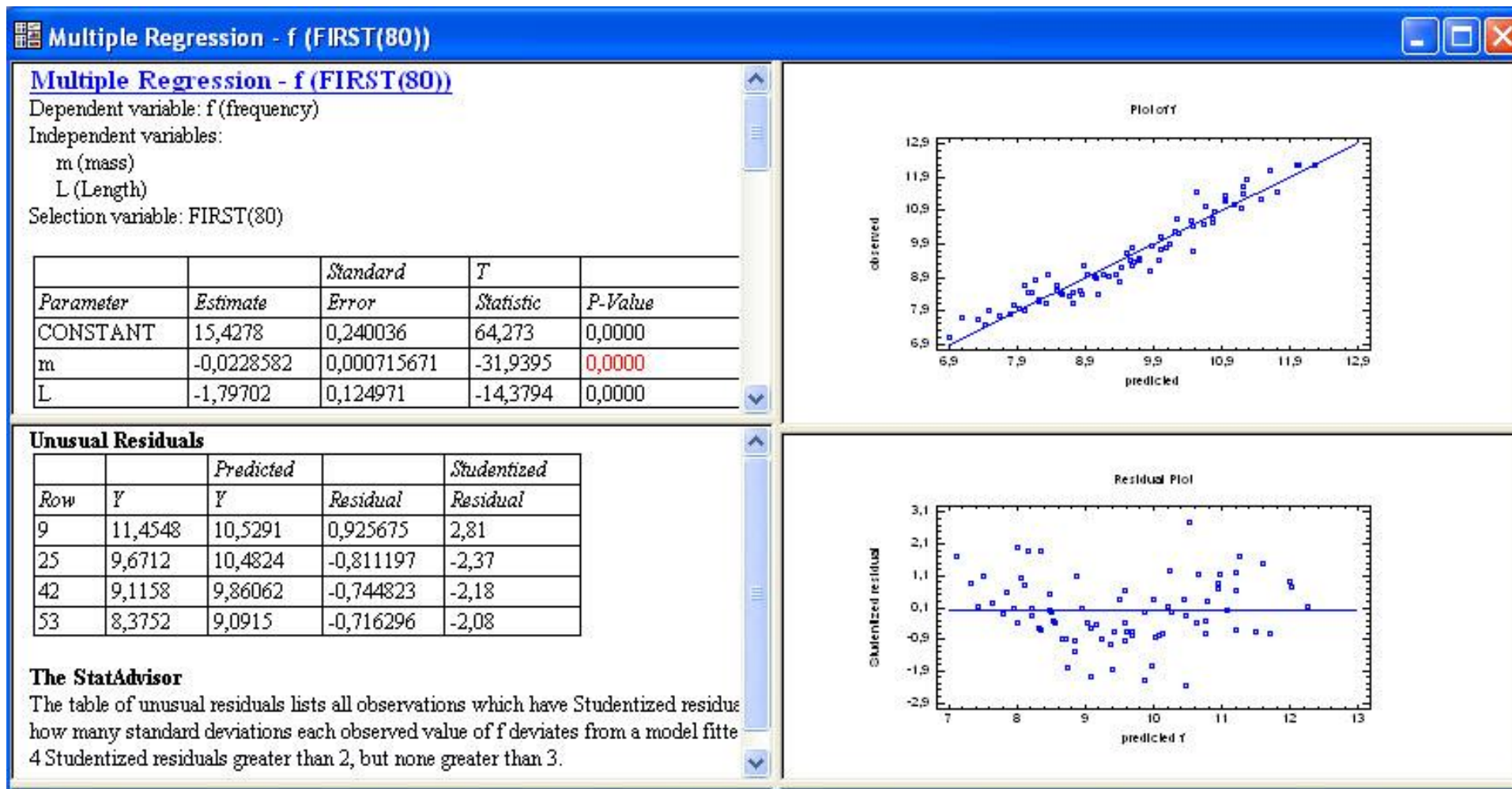


Рис. 3.12. Окно процедуры множественной регрессии

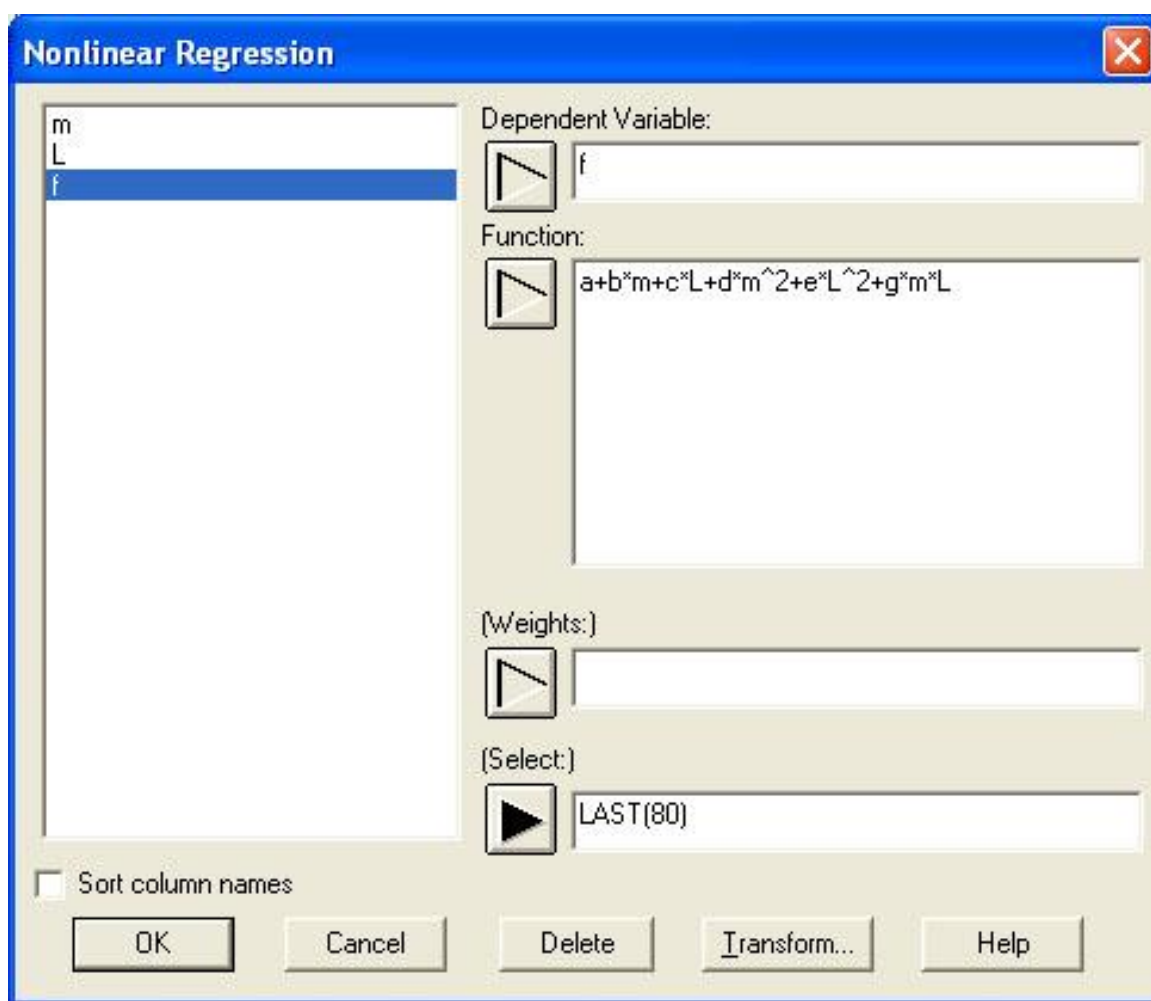


Рис. 3.13. Диалоговое окно ввода данных для нелинейной регрессии

В следующем окне (рис. 3.14) для каждой числовой константы установили ее предполагаемое первоначальное значение (*Initial Parameter Estimate*), изменяя которое программа по определенному алгоритму будет находить окончательное оптимальное значение.

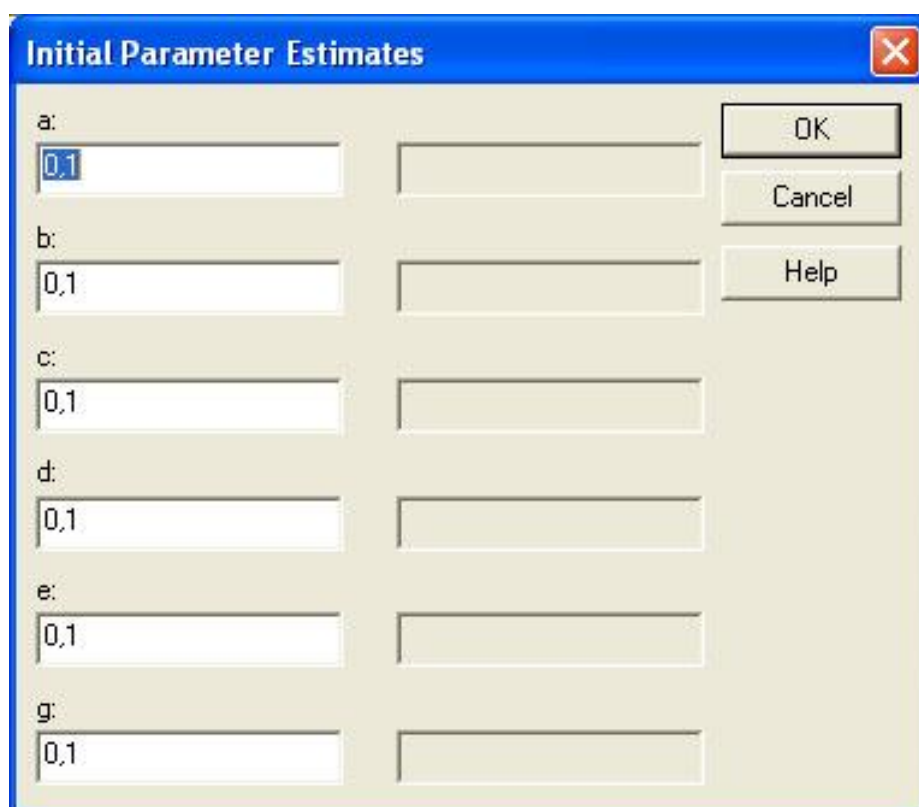
При первом запуске процедуры отыскания оптимальных значений задаются только эти коэффициенты (по умолчанию равны 0,1). Сам алгоритм (*Method*) и другие параметры поиска модели можно изменить в дальнейшем через пиктограмму *Analysis Options* (рис. 3.15).

Как видно из рис. 3.15, имеется возможность выбора одного из трех алгоритмов (по умолчанию – метод Марквардта, для которого также задаются дополнительные параметры). Кроме того, устанавливается уровень доверительной вероятности (*Confidence Level*) – статистическая мера числа

случаев, когда можно ожидать, что результаты эксперимента будут находиться внутри некоторого интервала (например, стандартный доверительный уровень 95 % означает, что результат действия будет соответствовать ожиданиям в 95 % случаев). Этот процент используется при вычислении асимптотических доверительных интервалов для коэффициентов модели. Алгоритм выбранного метода прекращает работу, когда соблюдено одно из условий, по которому эта остановка происходит. Эти условия прописываются в поля, ограниченные рамкой *Estimation* (Оценка). Достижение критериев остановки 1 или 2 (*Stopping Criterion 1* и 2) свидетельствует о том, что алгоритм сошелся и модель, удовлетворяющая всем условиям, найдена. Для удовлетворения первому критерию относительное изменение суммы квадратов остатков (*residuals*) между двумя итерациями должно быть меньше указанного значения. Для удовлетворения второму – относительное изменение по всем искомым коэффициентам должно быть меньше второго указанного значения. Достижение же максимального числа итераций (*Maximum Iterations*) или максимального числа вычислений функций (*Maximum Function Calls*), производимых несколько раз за одну итерацию, свидетельствует о том, что алгоритм не сошелся и модель, удовлетворяющая всем условиям, не найдена. Однако несхождение алгоритма не означает, что модель неприменима: так, при установке для рассматриваемой функции числа итераций, равного 3, алгоритм останавливается при достижении этого числа, при этом значение *R-квадрат* (94,7 %) будет вполне приемлемым, хотя это будет свидетельствовать о том, что при установке большего числа итераций достижима более качественная модель.

После закрытия диалогового окна задания предполагаемых первоначальных значений констант создается окно анализа. Отобрав графики и таблицы, наиболее интересные для исследования, получили окно, состоящее из пяти частей (рис. 3.16).

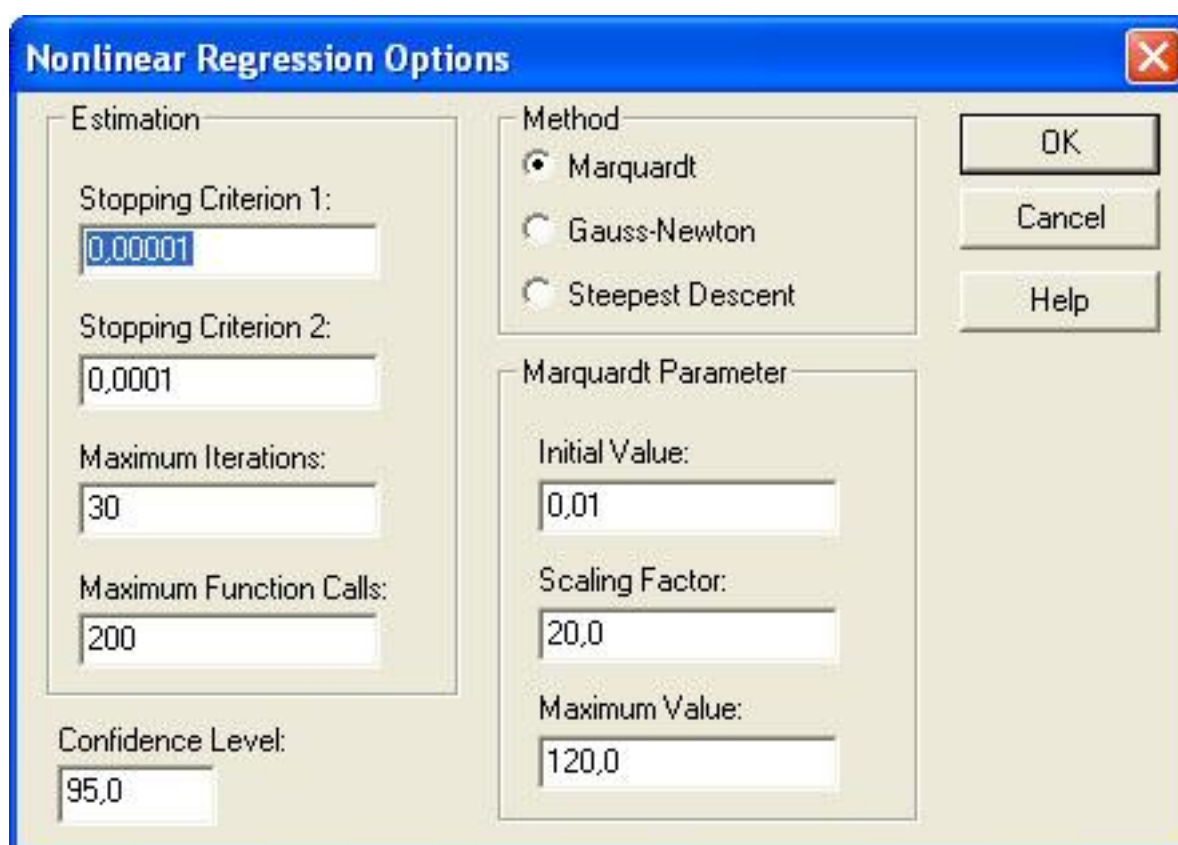
В левой верхней части представлена традиционная сводка анализа, в начале которой помещаются исходные данные анализа и информация



The dialog box titled "Initial Parameter Estimates" has a blue title bar with a close button (X) in the top right corner. It contains six rows of input fields for parameters a, b, c, d, e, and g. Each row has a label on the left, a text input field in the middle, and an empty text input field on the right. The values entered in the first input field of each row are: a: 0,1; b: 0,1; c: 0,1; d: 0,1; e: 0,1; g: 0,1. On the right side of the dialog, there are three buttons: "OK", "Cancel", and "Help", arranged vertically.

Parameter	Value
a:	0,1
b:	0,1
c:	0,1
d:	0,1
e:	0,1
g:	0,1

Рис. 3.14. Предполагаемые первоначальные значения констант



The dialog box titled "Nonlinear Regression Options" has a blue title bar with a close button (X) in the top right corner. It is divided into two main sections: "Estimation" on the left and "Method" on the right. The "Estimation" section contains four input fields: "Stopping Criterion 1:" (0,00001), "Stopping Criterion 2:" (0,0001), "Maximum Iterations:" (30), and "Maximum Function Calls:" (200). Below these is a "Confidence Level:" input field with the value 95,0. The "Method" section contains three radio buttons: "Marquardt" (selected), "Gauss-Newton", and "Steepest Descent". Below the radio buttons is a "Marquardt Parameter" section with three input fields: "Initial Value:" (0,01), "Scaling Factor:" (20,0), and "Maximum Value:" (120,0). On the right side of the dialog, there are three buttons: "OK", "Cancel", and "Help", arranged vertically.

Parameter	Value
Stopping Criterion 1:	0,00001
Stopping Criterion 2:	0,0001
Maximum Iterations:	30
Maximum Function Calls:	200
Confidence Level:	95,0
Method	Marquardt
Initial Value:	0,01
Scaling Factor:	20,0
Maximum Value:	120,0

Рис. 3.15. Параметры нелинейной регрессии

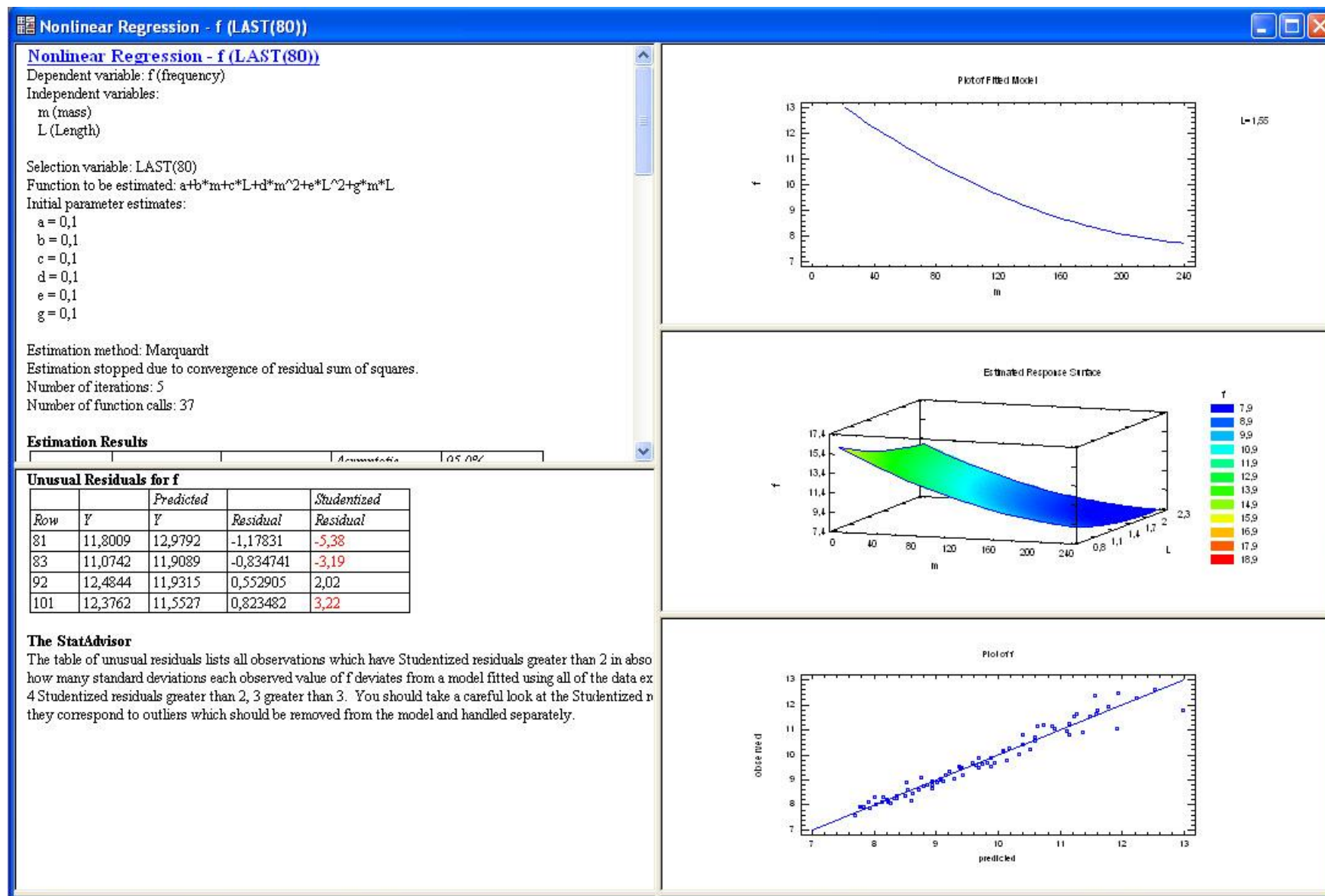


Рис. 3.16. Окно анализа нелинейной регрессии по полиномиальной функции

о его завершении (так, там сообщается, что алгоритм сошелся при 5 итерациях и 37 вычислениях функции). В левой нижней части расположена таблица «необычных остатков по частоте», в которой располагаются те наблюдения, для которых значение так называемого «студентизированного остатка» больше 2. Под остатком подразумевается разница между наблюдаемым значением и значением, предсказываемым моделью, а под студентизированным остатком – остаток, деленный на оценку своего стандартного отклонения, меняющегося от одного наблюдения к другому. Соответственно таблица содержит номер ряда (наблюдения), значение Y (частота), предсказываемое значение \hat{Y} , остаток и студентизированный остаток. Выделение точек в рядах 81, 83, 92 и 101 подтверждается графиком в правой нижней части (*Plot of f*), иллюстрирующим наложение наблюдаемых (*observed*) частот на модель предсказываемых (*predicted*) частот. На нем хорошо заметны 4 точки, расположенные дальше от прямой линии, чем другие. При этом 3 наблюдения, у которых студентизированный остаток больше 3, рекомендуется из модели удалять и рассматривать отдельно. В целом график демонстрирует вполне точное описание моделью реальных значений.

Влияние же отдельных факторов на частоту колебаний визуально отображается на двумерном графике построенной модели (*Plot of Fitted Model*), расположенном в правой верхней части, и трехмерном графике модели (*Estimated Response Surface*) в правой части в середине.

На первом из них отображается связь между частотой колебаний и массой груза при фиксированном значении длины каната (по умолчанию установлено среднее значение 1,55 м). Переключение на график отношения частоты и длины каната осуществляется через *Pane Options* (Опции текущего графика), где, помимо выбора связи, по изменяемому фактору можно задать нижний и верхний пределы, а по неизменяемому – фиксированное значение.

В результате рассмотрения нескольких графиков нашли подтверждение явно наблюдаемые в числовых данных тенденции: 1) масса груза оказывает большее влияние на частоту колебаний, чем длина каната; 2) чем меньше масса, тем больше частота колебаний; 3) чем меньше длина каната, тем больше частота колебаний; 4) самые низкие частоты колебаний с минимальным диапазоном наблюдаются при максимальной массе; 5) самые высокие частоты колебаний с минимальным диапазоном наблюдаются при минимальной массе; 6) колебания при максимальной длине каната охватывают весь диапазон частот.

Тенденции в двухмерных графиках модели нашли отражение в трехмерном графике в середине правой части. Его презентационное качество, хорошо иллюстрирующее модель, делает уместным размещение его в более выгодном свете, поэтому в окончательном варианте в левой части окна анализа была оставлена лишь аналитическая сводка (*Analysis Summary*), в полной мере описывающая построенную модель, а в правой части – ее трехмерное графическое представление (рис. 3.17).

Сводка анализа, помимо выше упомянутой информации о методе, его успешном схождении и числе итераций и вызовов функций, содержит 3 таблицы и итоговый анализ, сформированный *StatAdvisor*.

Первая таблица *Estimation Results* (Результаты оценки) содержит найденные значения константных параметров полиномиальной функции a , b , c , d , e и g (столбец *Estimate*), а также значения *Asymptotic Standard Error* (Асимптотической средней квадратической ошибки) и нижнего и верхнего пределов *Asymptotic 95 % Confidence Interval* (Асимптотического 95 % доверительного интервала). Во второй таблице *Analysis of Variance* (Дисперсионный анализ) размещена дисперсия (сумма квадратов) для модели, остатка и общая. После таблицы указываются, помимо прочего, значение показателя корреляции с экспериментальными данными *R-квадрат* (*R-square*),

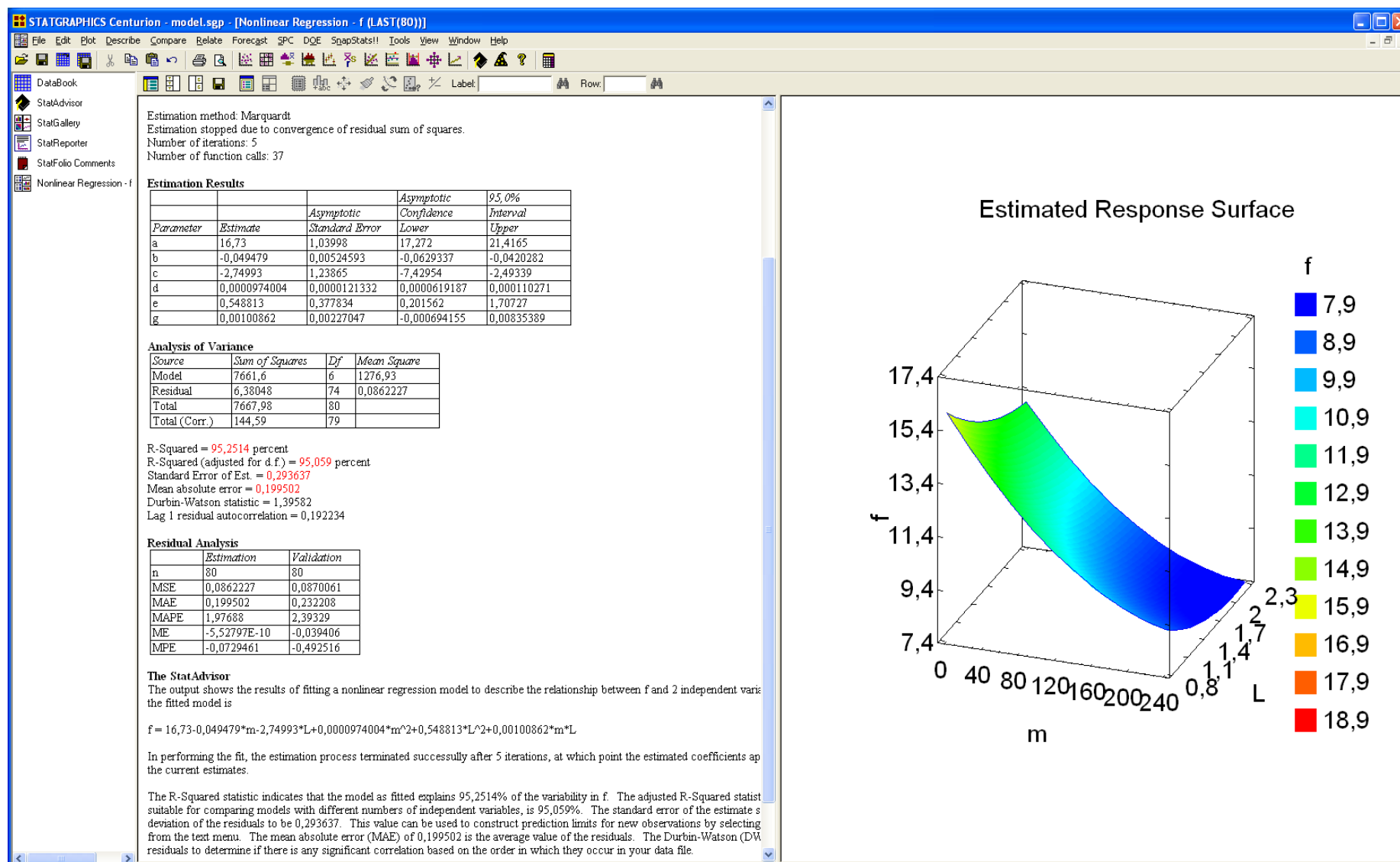


Рис. 3.17. Окно анализа нелинейной регрессии с двумя элементами

Таблица 3.2

Математические модели, описывающие зависимость f от m и L

Режим эксперимента	Номер функции	Аппроксимирующая функция $f = F(m, L)$	Показатель корреляции, R^2 , %
Опускание груза	1	$f = 14,1937 - 0,0219189 \cdot m - 1,18239 \cdot L$	91,8029
	2	$f = \exp(2,73458 - 0,00230794 \cdot m - 0,12325 \cdot L)$	92,9199
	3	$f = e^{3,59871} \cdot m^{-0,267948} \cdot L^{-0,191976}$	93,5932
	4	$f = 5,4562 + 236,398 \cdot \frac{1}{m} + 2,6001 \cdot \frac{1}{L}$	87,4658
	5	$f = 7,24576 + 238,031 \cdot \frac{1}{m + L}$	78,4802
	6	$f = 15,8225 - 0,0442935 \cdot m - 1,6236 \cdot L + 0,0000828595 \cdot m^2 + 0,121737 \cdot L^2 + 0,0000141647 \cdot m \cdot L$	95,0513
Подъем груза	7	$f = 13,8647 - 0,0221776 \cdot m - 0,909419 \cdot L$	90,6439
	8	$f = \exp(2,69264 - 0,00228459 \cdot m - 0,0940278 \cdot L)$	92,5732
	9	$f = e^{3,56753} \cdot m^{-0,264177} \cdot L^{-0,149803}$	94,7171
	10	$f = 5,87295 + 237,336 \cdot \frac{1}{m} + 2,01522 \cdot \frac{1}{L}$	89,4835
	11	$f = 7,36842 + 238,344 \cdot \frac{1}{m + L}$	83,5971
	12	$f = 16,73 - 0,049479 \cdot m - 2,74993 \cdot L + 0,0000974004 \cdot m^2 + 0,548813 \cdot L^2 + 0,00100862 \cdot m \cdot L$	95,2514
Подъем и опускание груза	13	$f = 14,059 - 0,02208 \cdot m - 1,06748 \cdot L$	91,1094
	14	$f = \exp(2,71674 - 0,00229918 \cdot m - 0,110967 \cdot L)$	92,6405
	15	$f = e^{3,58376} \cdot m^{-0,266274} \cdot L^{-0,171863}$	94,0575
	16	$f = 5,66761 + 237,085 \cdot \frac{1}{m} + 2,29054 \cdot \frac{1}{L}$	88,4242
	17	$f = 7,30352 + 238,574 \cdot \frac{1}{m + L}$	80,9701
	18	$f = 15,8894 - 0,0467482 \cdot m - 1,60954 \cdot L + 0,0000915177 \cdot m^2 + 0,145581 \cdot L^2 + 0,0000618428 \cdot m \cdot L$	95,0248

равное 95,2514 %, средняя квадратичная ошибка оценки (*Standard Error of Estimate*) – 0,293637, средняя абсолютная ошибка (*Mean absolute error*) – 0,199502. В третьей таблице располагается анализ остатка (*Residual Analysis*), представляющий собой расчет пяти видов средних ошибок (квадратичной, абсолютной, абсолютной в процентах, простой и простой в процентах) – отклонений восьмидесяти наблюдаемых частот колебаний от значений частот, предсказываемых по модели (*Estimation*).

Итоговый анализ *StatAdvisor* начинается с математического представления следующей построенной модели:

$$f = 16,73 - 0,049479m - 2,74993L + 0,0000974004m^2 + 0,548813L^2 + 0,00100862mL.$$

Затем сообщает о том, что процесс анализа закончился успешно после 5 итераций, обращает внимание на то, что найденное значение *R*-квадрат показывает, что модель объясняет 95,2514% изменений частоты колебаний, дает прочие рекомендации по интерпретации результатов и прогнозированию.

Подобным образом было получено 18 математических моделей, описывающих зависимость частоты колебаний при торможении от массы груза и длины каната [26]. Причем в каждом случае путем варьирования параметров анализа находилось наибольшее значение показателя корреляции *R*-квадрат расчетных данных с экспериментальными (табл. 3.2).

Как видно из таблицы, наиболее точно описывают связь частоты колебаний груза с его массой и длиной каната модели, использующие степенную (номера 3, 9 и 15) и полиномиальную (номера 6, 12 и 18) функции. Причем это имеет место как в случае подъема груза, так в случае опускания. А поскольку значение показателя корреляции с экспериментальными данными очень высокое (94–95 %), устройства, устраняющие колебания, в частности описанные в гл. 2, вполне можно совершенствовать на основе этих моделей.

4. МОДЕРНИЗАЦИЯ СУЩЕСТВУЮЩИХ СРЕДСТВ УСТРАНЕНИЯ ВЕРТИКАЛЬНЫХ КОЛЕБАНИЙ ГРУЗОВ НА ОСНОВЕ ПРОВЕДЕННОГО ИССЛЕДОВАНИЯ

4.1. Модернизация магнитореологического виброгасителя

Используя математические модели, полученные выше, наилучшие из устройств, описанных во 2-й главе, можно сделать еще более совершенными. Рассмотрим, в частности, магнитореологический виброгаситель, показанный на рис. 2.9. При всех своих положительных качествах у него имеется существенный недостаток: датчик частоты 8 и устройство управления 9 установлены в нем непосредственно на корпусе 1. Датчик выдает сигнал довольно сложного спектра. Для того чтобы надежно управлять электромагнитом, избежать неоправданных потерь энергии в нем, из сигнала нужно выделить основную гармонику. Это требует применения в блоке управления системы частотных фильтров и соответствующих виброусилителей. Если это октавные фильтры (как это имеет место в аппаратуре ВИБ-6ТН), то их требуется восемь, и, соответственно, может потребоваться восемь усилителей. Все это вместе с блоком питания делает устройство управления достаточно громоздким. Установка же такого устройства на корпусе виброгасителя, который, в свою очередь, крепится на крюковой подвеске, подвергает его значительной опасности повреждений и поломок. Снизить опасность повреждений можно путем размещения устройства управления вдали от крюковой подвески, однако в этом случае его нужно будет связывать проводами с датчиком, оставшимся на ней, что вряд ли будет способствовать повышению надежности виброгасителя. Между тем, если применения вибродатчика в рассматриваемом виброгасителе избежать, то устройство управления вдали от крюковой подвески (например, в кабине машиниста крана) разместить станет возможно. Однако для этого частоту вынужденных колебаний подвески с грузом придется определять

иным способом, например вычислять по одной из формул, приведенных в табл. 3.2. Это потребует измерения в процессе работы крана величин m и L , но, учитывая то, что современные краны, как правило, оснащаются подобными измерителями в штатном порядке, такое измерение труда не составит.

Из всех формул, приведенных в табл. 3.2, наиболее точно зависимость f от m и L при подъеме и опускании груза краном описывает формула под номером 18. На ее основе и целесообразно произвести модернизацию магнитореологического виброгасителя.

Схема модернизированного магнитореологического виброгасителя показана на рис. 4.1. Виброгаситель имеет корпус 1, установленный на крюковой подвеске 2 подъемно-транспортной машины. Внутри корпуса 1 размещены электромагнит 3 и магнитореологическая суспензия 4. Имеется также груз 5, прикрепленный к корпусу 1 посредством пружин, выполненных в виде упругих стержней 6. Груз 5 (он может быть изготовлен в виде цельного кольца или отдельных частей с дросселирующими отверстиями) размещается в суспензии 4. При необходимости он может быть составным, таким, что его часть 7 может быть размещена вне корпуса 1 на наконечниках стержней 6 и установлена на резьбе. Стержни 6 также являются частью груза 5. Помимо перечисленного, гаситель включает в себя датчик 8 массы объекта транспортирования, датчик 9 длины каната, на котором закреплена крюковая подвеска 2, первый 10 и второй 11 усилители с регулируемыми коэффициентами усиления, задатчик напряжения 12 и двухвходовой сумматор напряжений 13 и компаратор 14. Датчик массы 8 соединен с входом усилителя 10, датчик длины каната 9 соединен с входом усилителя 11, выход усилителя 10 соединен с первым входом сумматора 13, выход усилителя 11 соединен со вторым входом сумматора 13, выход сумматора 13 соединен с шиной ввода в компаратор 14 вычитаемого сигнала, выход задатчика 12 соединен с шиной ввода в компаратор 14 уменьшаемого сигнала. Кроме того, гаситель снабжен первым 15, вторым 16 и

третьим 17 двухходовыми блоками умножения сигналов, третьим 18, четвертым 19 и пятым 20 усилителями с регулируемыми коэффициентами усиления и четырехходовым сумматором 21. Выход датчика массы 8 соединен с первым и вторым входами блока 15 и первым входом блока 16, выход датчика длины каната 9 соединен с первым и вторым входами блока 17 и вторым входом блока 16, выход блока 15 соединен со входом усилителя 18, выход блока 16 соединен со входом усилителя 19, выход блока 17 соединен со входом усилителя 20, выходы усилителей 18, 19, 20 соответственно соединены с первым, вторым и третьим входами четырехходового сумматора 21, выход компаратора 14 соединен с четвертым входом сумматора 21, а выход последнего связан с электромагнитом 3.

Работа предлагаемого виброгасителя основана на зависимости

$$f = A - (Bm + CL) + Dm^2 + EL^2 + KmL,$$

где m – масса объекта транспортирования; L – длина каната, на котором закреплена крюковая подвеска; A, B, C, D, E, K – константы, характеризующие упругие свойства каната; f – частота вынужденных колебаний объекта транспортирования и подвески на канате длины L .

Перед использованием виброгасителя его вначале настраивают. Для этого, ориентируясь на объект транспортирования некоторой средней массы m , перемещаемой краном на высоте, соответствующей средней длине каната L , в зависимости от марки примененного каната выбирают значения констант A, B, C, D, E, K . С помощью задатчика 12 в компаратор вводят напряжение, соответствующее A , затем коэффициенты усиления первого и второго усилителей 10 и 11 принимают равными соответственно B и C и настраивают усилители на эти коэффициенты. Аналогично коэффициенты усиления усилителей 18, 20 и 19 принимают равными соответственно D, E и K и настраивают эти усилители на эти коэффициенты.

Далее устанавливают на крановой подвеске объект средней массы m , поднимают его на высоту, соответствующую средней длине каната L , и тормозят. Возникают вертикальные колебания. Повторяя это несколько

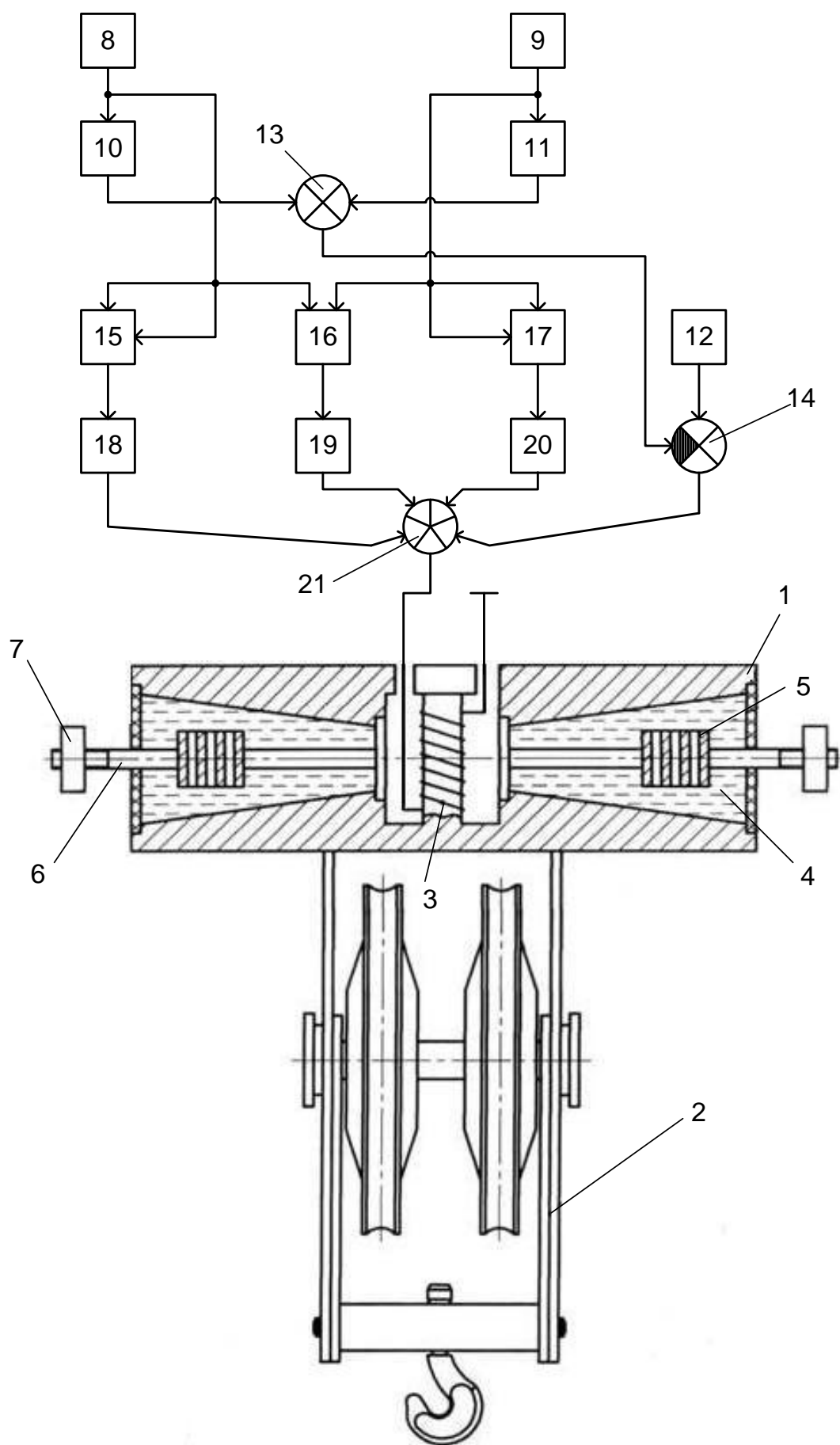


Рис. 4.1. Модернизированный магнитореологический гаситель колебаний

раз, регулируют положение грузов и корректируют A , B , C , D , E , K так, чтобы колебания затухали как можно быстрее. Виброгаситель настроен. Теперь его можно эксплуатировать, не перенастраивая при транспортировке любых грузов (перенастройка потребуется только при замене каната).

При дальнейшей эксплуатации происходит автоматическое определение массы m и длины L , также автоматическое определение величин Bm и CL , Dm^2 , EL^2 и KmL , частоты f и ее отображение на выходе сумматора 21.

Это напряжение подается на электромагнит 3, он, в свою очередь, создает магнитное поле, заставляющее магнитореологическую суспензию 4 приобрести вязкость, при которой демпфирующая способность виброгасителя будет близка к частоте колебаний объекта транспортирования и крюковой подвески. В результате колебания объекта с подвеской быстро затухнут.

Поскольку модернизированный виброгаситель не имеет вибродатчика, устанавливаемого непосредственно на его корпусе, то устройство определения f и блок его питания, как уже отмечалось, могут быть размещены не на крюковой подвеске, а в кабине машиниста крана. Конечно, при этом потребуется провод, соединяющий указанное устройство с электромагнитом, но это вряд ли является серьезным недостатком модернизации.

Описанный виброгаситель предполагает выполнение на аналоговых элементах. Однако он может быть изготовлен и в дискретном варианте. Сегодня датчики массы груза и длины каната зачастую выполняются цифровыми, поэтому блок определения f виброгасителя может быть реализован на микропроцессоре, выход которого связывается с электромагнитом через ЦАП.

4.2. Модернизация виброгасителя с автоматическим управлением статической осадкой груза на мембране

Недостаток, присущий магнитореологическому виброгасителю, описанному в гл. 2, свойственен и другим существующим устройствам актив-

ного виброгашения, в том числе и виброгасителю, схема которого приведена на рис. 2.1. В данном случае устранить его можно так же, как и в предыдущем случае. Вместе с тем такой виброгаситель имеет и еще один существенный недостаток: сигнал, управляющий статической осадкой X груза на мембране, в нем пропорционален частоте f вынужденных колебаний. Между тем известно [27], что частота собственных колебаний f_c виброгасителя связана со статической осадкой X груза обратной пропорциональной зависимостью

$$f_c^2 = \frac{R}{X},$$

где R – некоторый коэффициент пропорциональности, определяемый конструктивными особенностями виброгасителя. Отсюда следует, что при работе рассматриваемого виброгасителя его автоматическая поднастройка в резонанс происходит лишь приближенно. Для того чтобы поднастройка происходила более точно, блоки определения f , примененные в магнито-реологическом виброгасителе, в данном случае необходимо дополнить еще двумя: еще одним двухвходовым блоком умножения сигналов и блоком деления. Как это осуществлено, показано на рис. 4.2.

Виброгаситель имеет шток 1, устанавливаемый на крюковой подвеске 2 крана. На штоке перпендикулярно его оси закреплена упругая мембрана 3, на которой подвешен груз 4. На штоке же над грузом 4 неподвижно закреплен электромагнит 5. Вход электромагнита соединен с выходом блока деления 6, а вход блока деления подключен к выходу дополнительного двухвходового блока умножения сигналов 7. Остальные блоки рассматриваемого виброгасителя идентичны блокам предыдущего, с той лишь разницей, что выход сумматора 21 в нем соединен не напрямую с электромагнитом, а со входами блока 7 и подключен к электромагниту через него и блок 6.

Перед началом эксплуатации виброгасителя, так же как и в предыдущем случае, в него с помощью задатчика 12 вводят константу A и усили-

тели 10, 11, 18, 20 и 19 настраивают, устанавливая их коэффициенты усиления соответственно равными B , C , D , E , K . Далее настраивают делитель 6 на коэффициент деления R . С помощью груза некоторой средней массы, поднимаемого краном на высоту, соответствующую некоторой средней длине каната, величины A , B , C , D , E , K , R уточняют подобно тому, как это делалось при использовании магнитореологического виброгасителя. После этого виброгасителем можно пользоваться при подъеме и опускании краном разных грузов на разную высоту. Если частота вынужденных колебаний груза с крюковой подвеской 2 относительно некоторой средней, соответствующей средним m и L , увеличится, то на выходе блока 7 увеличится квадрат частоты вынужденных колебаний, а на выходе 6 – величина f^2/R . Это приведет к соответствующему уменьшению статической осадки X груза 4 на мембране 3. Однако уменьшение X приведет к соответствующему увеличению f_c^2/R (напомним, что f_c – частота собственных колебаний виброгасителя), и произойдет поднастройка виброгасителя в резонанс. Если частота вынужденных колебаний уменьшится, то X увеличится и все произойдет аналогично. При этом точность поднастройки в резонанс будет, естественно, выше, чем в таком же виброгасителе, но немодернизированном.

4.3. Модернизация системы двухступенчатого торможения с определением полупериода вынужденных колебаний на основе измерения длины каната и массы груза, перемещаемого краном

Подход к определению f путем вычисления на основе измерения m и L может быть использован не только при усовершенствовании устройств активного виброгашения, как это было сделано выше, но и при модернизации рассмотренных также в гл. 2 систем двухступенчатого торможения,

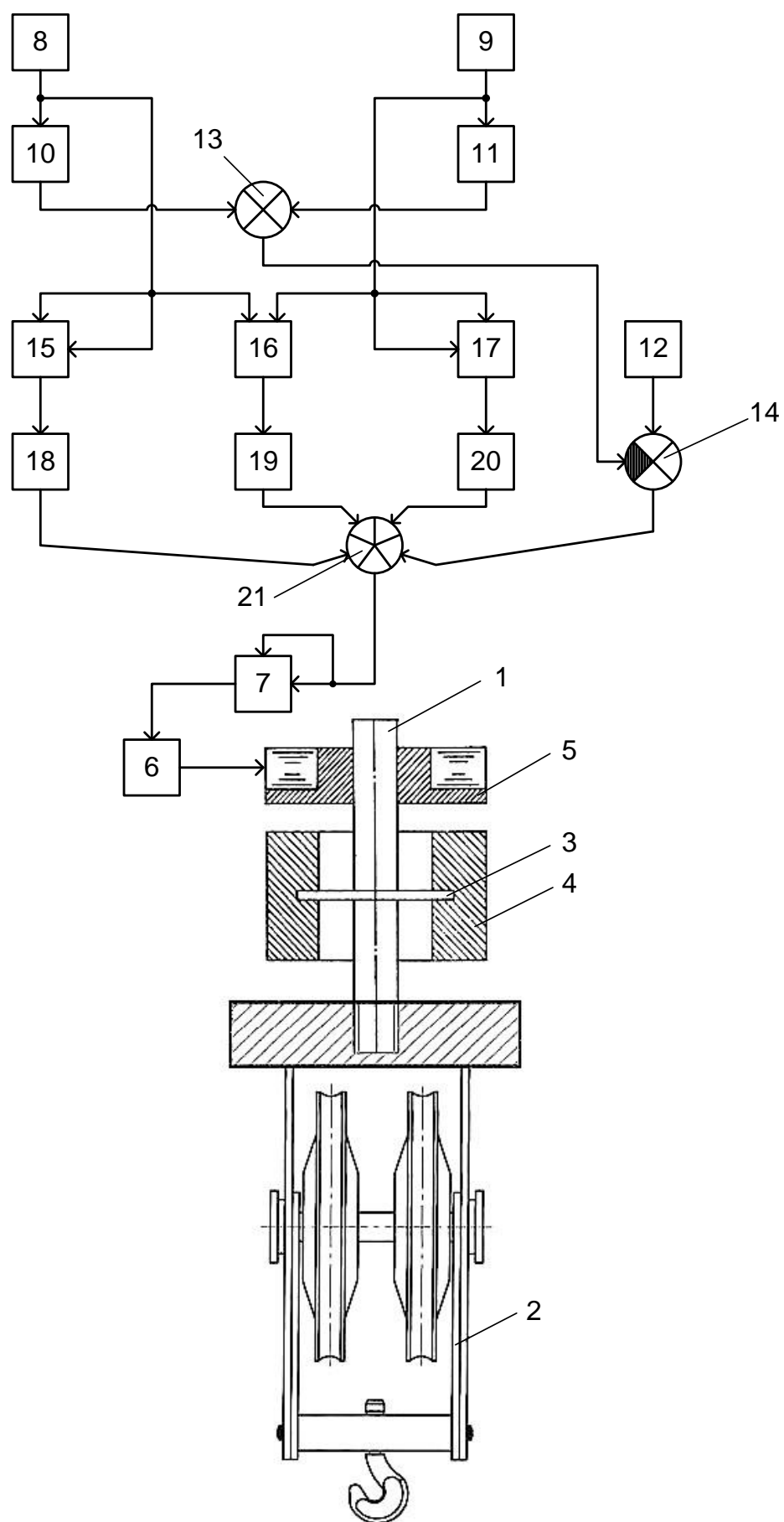


Рис. 4.2. Модернизированный виброгаситель с управлением статической осадкой груза

в частности системы, защищенной патентом РФ № 87419.

Как отмечалось, величина полупериода колебаний $T/2$ при построении этой системы в соответствии с формулой (9) была принята равной

$$T/2 = a + bL + ct,$$

где a, b, c – некоторые действительные числа. Формула получена, по сути, интуитивно, а потому, естественно, является весьма приближительной. Если, зная, что $T/2 = 1/2f$, использовать для определения $T/2$ модель номер 18 из табл. 3.2, то точность определения полупериода $T/2$, как очевидно, будет выше.

Для того чтобы использовать указанную модель, в систему по патенту № 87419 вместо подсистемы вычисления $T/2$, состоящей из блоков 1–5 (рис. 2.17), была включена подсистема, состоящая из блоков 8–21 (рис. 4.4). Выход этой подсистемы (выход сумматора 21) был соединен с дополнительным блоком 6 – блоком деления числа $1/2$ на входной сигнал. Поскольку последний равен f , то на выходе блока 6 как раз и получается $T/2$.

Модернизируя систему двухступенчатого торможения по патенту № 87419, попутно можно решить еще одну задачу, связанную с повышением эффективности гашения колебаний. Как математически обосновано в работе [28], динамические нагрузки, возникающие при двухступенчатом торможении подъемной машины, можно существенно уменьшить, если тормоза первой и второй ступеней включать согласно графику, приведенному на рис. 4.3: вначале в момент времени $t = 0$ включить первую ступень торможения, затем в момент времени t_A (по истечении времени $T/2$) включить вторую ступень, далее в момент времени t_B , когда скорость подъема уменьшится до величины

$$V_{\text{гр}} = \frac{F_{\text{т}} / 2}{m_{\Sigma}} T / 2, \quad (*)$$

(F_T – тормозные усилия обеих ступеней, m_Σ – масса движущихся частей машины), вторую ступень выключить, а по истечении времени $T/2$ в момент времени t_C снова включить. Как показывают исследования, проведенные в Пермском государственном университете под руководством А. Г. Степанова, при снижении величины $V_{гр}$, груз, закрепленный на крюковой подвеске, находится в крайней точке колебаний подвески. При снятии половины тормозного усилия груз за время $T/2$ достигает положения, соответствующего действию на канат только статической нагрузки. Если в этот момент произвести стопорение машины, то колебания практически сразу же прекратятся.

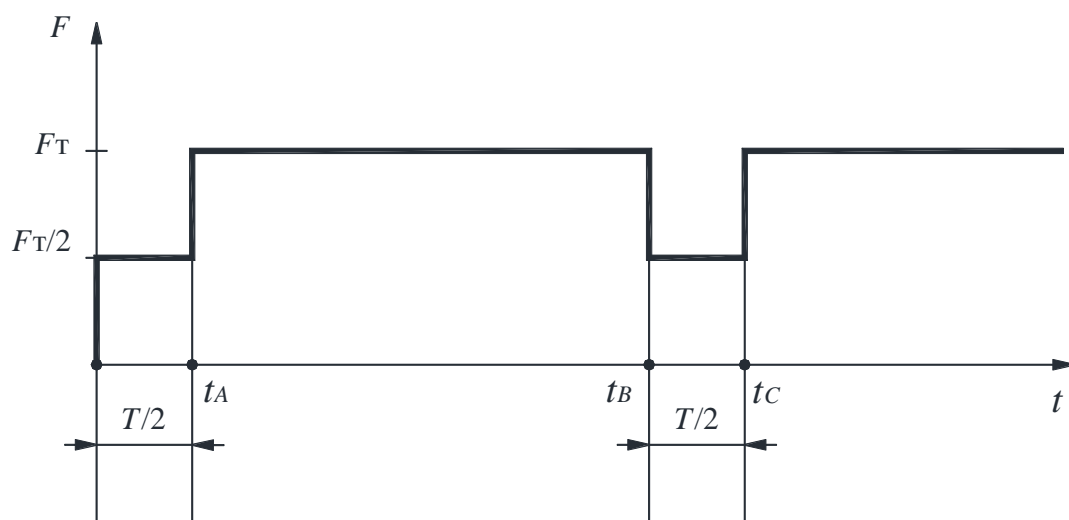


Рис. 4.3. График включения – выключения двух ступеней торможения

Схема системы по патенту № 87419, модернизированной согласно вышеизложенному, изображена на рис. 4.4. На нем следующие обозначения: 1 – тормоз первой ступени; 2 – тормоз второй ступени (по тормозному моменту они равны); программируемое реле времени 3; регулируемый пороговый элемент 4 аварийного отключения привода подъема груза; датчик скорости 5 подъема груза; блок 6 деления числа $1/2$ на f ; блок 7 – задатчик $F_T/2$ и m' – величины массы движущихся при подъеме частей машины (сюда не входит масса подвески с грузом); блоки 8 – 21 определения f (те

же, что и у виброгасителей по рис. 4.1 и 4.2); инвертор 22; первый элемент дифференцирования 23; дополнительный регулируемый пороговый элемент 24; *RS*-триггер 25; второй элемент дифференцирования 26; двухходовой дизъюнктор 27; дополнительный сумматор 28, первый вход которого выполнен смещающим, а второй выполнен управляющим; блок 29 деления числа $F_T/2$ на m_Σ , первый вход которого выполнен в виде шины ввода сигнала, являющегося делимым, а второй вход – в виде шины ввода сигнала, являющегося делителем; блок умножения 30.

Элементы и блоки системы соединены следующим образом: тормоз первой ступени 1 своей цепью управления соединен с элементом дифференцирования 26 и с шиной подачи сигнала *R* включения системы; тормоз второй ступени 2 своей цепью управления соединен с выходом *RS*-триггера 25 (выходом служит одна «ножка» триггера; вторая остается свободной); реле времени 3 своим программирующим входом соединено с выходом блока деления 6; вход порогового элемента 4 отключения привода подъема груза соединен с выходом датчика массы 8.

Выход датчика массы 8 соединен со входом усилителя 10 и с первым входом сумматора 28; выход датчика скорости 5 соединен с рабочим входом порогового элемента 24, выход которого через инвертор 22 подключен ко входу элемента дифференцирования 23; выход элемента 23 соединен со вторым входом *RS*-триггера 25 и с первым входом дизъюнктора 27.

Первый вход *RS*-триггера соединен с выходом реле времени 3; выход второго элемента дифференцирования 26 соединен со вторым входом дизъюнктора 27, выход которого подключен к управляющему входу реле времени 3; выход сумматора 28 соединен со входом блока деления 29; выход блока 29 соединен со вторым входом блока умножения 30, первый вход которого связан с выходом блока деления 6, а выход подключен к регулировочному входу порогового элемента 24.

Перед использованием системы ее вначале настраивают. По результатам испытания каната определяют соответствующие ему значения A, B, C, D, E, K и вводят их в подсистему определения f . В зависимости от допустимой перегрузки машины задают порог срабатывания порогового элемента 4. С помощью смещающего входа сумматора 28 его настраивают, вводя в него величину m' . В блок деления 29 вводят делимое, равное $F_T / 2$.

При дальнейшей эксплуатации машины уже ничего настраивать не требуется. Нужно лишь сбросить на «ноль» RS-триггер 25. Когда машина поднимает груз, масса которого вместе с крюковой подвеской равна m , и тормозится на высоте подъема L , на выходе блока 6 появляется сигнал $T / 2$. Он поступает на программирующий вход реле времени 3 и задает его время срабатывания. Одновременно с этим сигнал m поступает от датчика 8 на сумматор 28, складывается в нем с величиной m' и подается на блок деления 29. На выходе этого блока появляется сигнал $\frac{F_T}{2(m + m')}$, который далее подается на блок умножения 30. На этот блок подается также сигнал $T / 2$ с выхода блока 6. В результате на выходе блока 30 оказывается сигнал $V_{гр}$, вычисленный по формуле (*). Этим сигналом устанавливается порог срабатывания порогового элемента 24.

Когда машину требуется затормозить, на шину R подается сигнал. Он включает тормоз первой ступени 1 и поступает на элемент дифференцирования 26. На выходе этого элемента появляется короткий импульс, который проходит через дизъюнктор 27 и поступает на управляющий вход реле времени 3. Реле начинает работать и отсчитывать время.

По истечении запрограммированного в нем времени $T / 2$ оно срабатывает, подает импульс на первый вход триггера 25 и переключает его. На выходе триггера появляется сигнал, включающий тормоз второй ступени 2. В результате торможения скорость перемещения груза уменьшается. Датчик скорости 5 выдает текущий сигнал скорости V

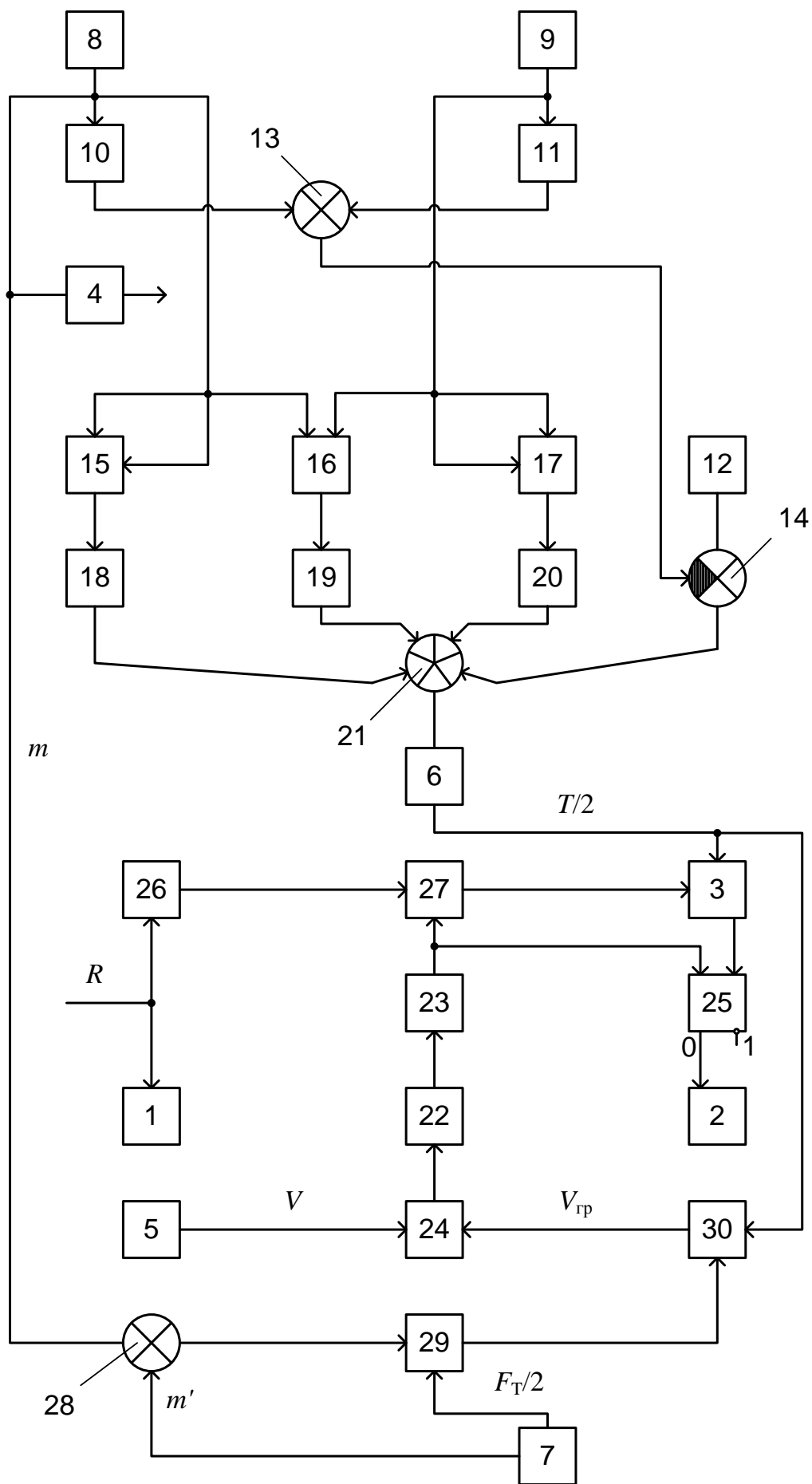


Рис. 4.4. Модернизированная система двухступенчатого торможения

на рабочий вход порогового элемента 24, где сравнивается с порогом его срабатывания $V_{гр}$. Пока $V > V_{гр}$ на выходе элемента 24 существует логический сигнал «1», преобразуемый инвертором 22 в логический «0». Как только будет иметь место $V \leq V_{гр}$, на выходе порогового элемента 24 появится «0», на выходе инвертора 22 появится «1». Этот сигнал поступит на элемент дифференцирования 23, а тот выдаст импульс на триггер 25, переключит его и выключит тормоз второй ступени 2. Одновременно с этим тот же импульс, пройдя через дизъюнктор 27, поступит на управляющий вход реле времени 3, а оно снова начнет отсчитывать запрограммированное время $T/2$. Как только отсчитает, выдаст сигнал на триггер 25, переключит его, а тот включит тормоз второй ступени 2 снова. Таким образом, при работе системы будет реализован график изменения тормозного усилия, приведенный на рис. 2, что приведет к ускоренному прекращению продольных колебаний каната с грузом.

Во всех рассмотренных устройствах блок определения f может быть не только принципиально, но и конструктивно выполнен одинаково. Это позволяет изготавливать его как унифицированный элемент указанных устройств. В результате описанные усовершенствования не окажутся столь уж сложными. Вместе с тем их применение не исключает использования для устранения колебаний и контроллеров плавного разгона – торможения приводов подъема–опускания грузов, описанных в п 2.3.

4.4. Компьютерная система плавного разгона–торможения привода подъема–опускания груза, перемещаемого краном

Как отмечалось в п. 2.3, устранение продольных колебаний каната с закрепленным на нем грузом может осуществляться и с помощью устройств управления крановыми приводами, обеспечивающих более или менее плавные разгон и торможение приводов. Наиболее перспективными из таких устройств, как указано в п. 2.3, являются электронные контроллеры с

реверсивным счетчиком импульсов и ЦАП, используемые в сочетании со следящими приводами. Они позволяют изменять управляющий сигнал, подаваемый на привод, при пуске и остановке последнего, аналогично показанному на рис. 2.23, в результате чего динамические нагрузки, вызывающие колебания, уменьшаются. Однако существенным недостатком контроллеров с реверсивным счетчиком является то, что и разгон, и торможение следящего привода они обеспечивают с постоянным ускорением – по линейному закону. В начале и при завершении разгона и торможения из-за этого зачастую имеют место значительные скачки ускорения, а значит, и сил, вызывающих колебания. И хотя колебания, о которых идет речь, при использовании контроллеров с реверсивным счетчиком оказываются меньше, чем при использовании контроллеров иных типов, они все же полностью не устраняются. Между тем известно, что колебания перемещаемой массы можно практически полностью предотвратить, если разгон и торможение перемещающего ее привода производить не по линейному закону, а по закону кубической параболы [29].

С помощью контроллеров на реверсивном счетчике формирование сигнала, управляющего разгоном–торможением следящего привода по закону кубической параболы, либо неосуществимо, либо возможно за счет существенного усложнения контроллера. Вместе с тем это вполне реализуемо без значительных затрат, если вместо электронного контроллера указанного типа использовать программируемый контроллер типа управляющей микро-ЭВМ. Такие контроллеры в последние годы в связи с повышением их технологичности и уменьшением стоимости в составе крановой автоматики применяются все шире, а потому, если их использовать для управления разгоном – торможением привода подъема–опускания грузов, перемещаемых краном, то можно получить еще более перспективный контроллер, чем построенный на реверсивном счетчике. Но применение программируемого контроллера типа микро-ЭВМ невозможно без специ-

ального программного обеспечения. В связи с этим такое обеспечение потребовалось разработать.

Разработанное программное обеспечение представляет собой систему плавного разгона–торможения привода подъема–опускания груза, перемещаемого краном, в основу которой положена программная реализация на языке C++ алгоритма, приведенного на рис. 4.5.

Данный алгоритм базируется на принципе разгона–торможения привода в соответствии с уравнением кубической параболы с аппроксимацией кривых отрезками прямых, параметры которых заложены таким образом, чтобы обеспечить наиболее плавные процессы исходя из имеющихся практических и теоретических соображений.

Перед использованием системы в нее вводятся значения времени, скорости и ускорения для восьми участков разгона и торможения (V_i, t_i, a_i , где $i = 1 \dots N \dots M$ – номер участка; $N = 3$ – номер последнего участка на стадии разгона; $M = 8$ – номер последнего участка во всем процессе разгона – торможения). Эти значения рассчитываются системой исходя из пяти введенных оператором параметров:

- *времени разгона (t_p)* – времени, за которое привод разгоняется до рабочей скорости;
- *рабочей скорости (V_p)* – скорости, на которой происходит рабочий процесс;
- *времени торможения (t_T)* – времени снижения скорости привода с рабочей скорости до нуля;
- *времени замедления (t_z)* – времени работы привода на замедленной скорости;
- *замедленной скорости (V_z)* – скорости, на которой привод будет работать время t_z перед окончательной остановкой.

Еще один параметр T – период вывода значения скорости через контроллер на управляемый привод – непосредственно оператором не задает-

ся и по умолчанию устанавливается равным 0.0001 с. Однако при необходимости он может быть изменен в соответствующем файле настроек программы. Необходимость изменения этого параметра может быть обусловлена наличием проблем при работе с конкретным приводом, если он некорректно обрабатывает высокочастотные сигналы. Но это бывает редко, и в общем случае изменения T не требуется.

Для удобства в алгоритме задаются также нулевые значения скорости, времени и ускорения ($V_0 = 0$, $t_0 = 0$, $a_0 = 1$). Указанная величина V_0 используется только при описании 1-го участка, t_0 – при старте (как t_i) и в дальнейшем при расчете промежутка времени между циклами Δt ($\Delta t = t - t_0$, где t – текущее время, а t_0 – предыдущее время), a_0 приравнено к 1 для обеспечения универсальности алгоритма при первоначальном выводе скорости на привод (где $a_i \neq 0$, т. е. скорость должна быть измененной).

После ввода и установки указанных выше значений система производит расчет параметров для каждого участка.

Отдельные участки кривой выбираются из следующих соображений: первый – движение с малым ускорением для выбора зазоров в кинематической цепи, второй – максимально допустимое ускорение/замедление при формировании переходных процессов, третий – снижение ускорения перед выходом на рабочую скорость или остановом. Таким образом, и стадия разгона, и стадия торможения изначально разбиваются на 3 части. При этом при разработке алгоритма было решено сделать эти части фиксированными с целью упрощения пользования системой и для ее универсальности.

В результате программа делит общее изменение скорости до рабочей на 3 части в соотношении 1/6, 4/6, 1/6, а время – в соотношении 2/5, 1/5, 2/5 (рис. 4.6).

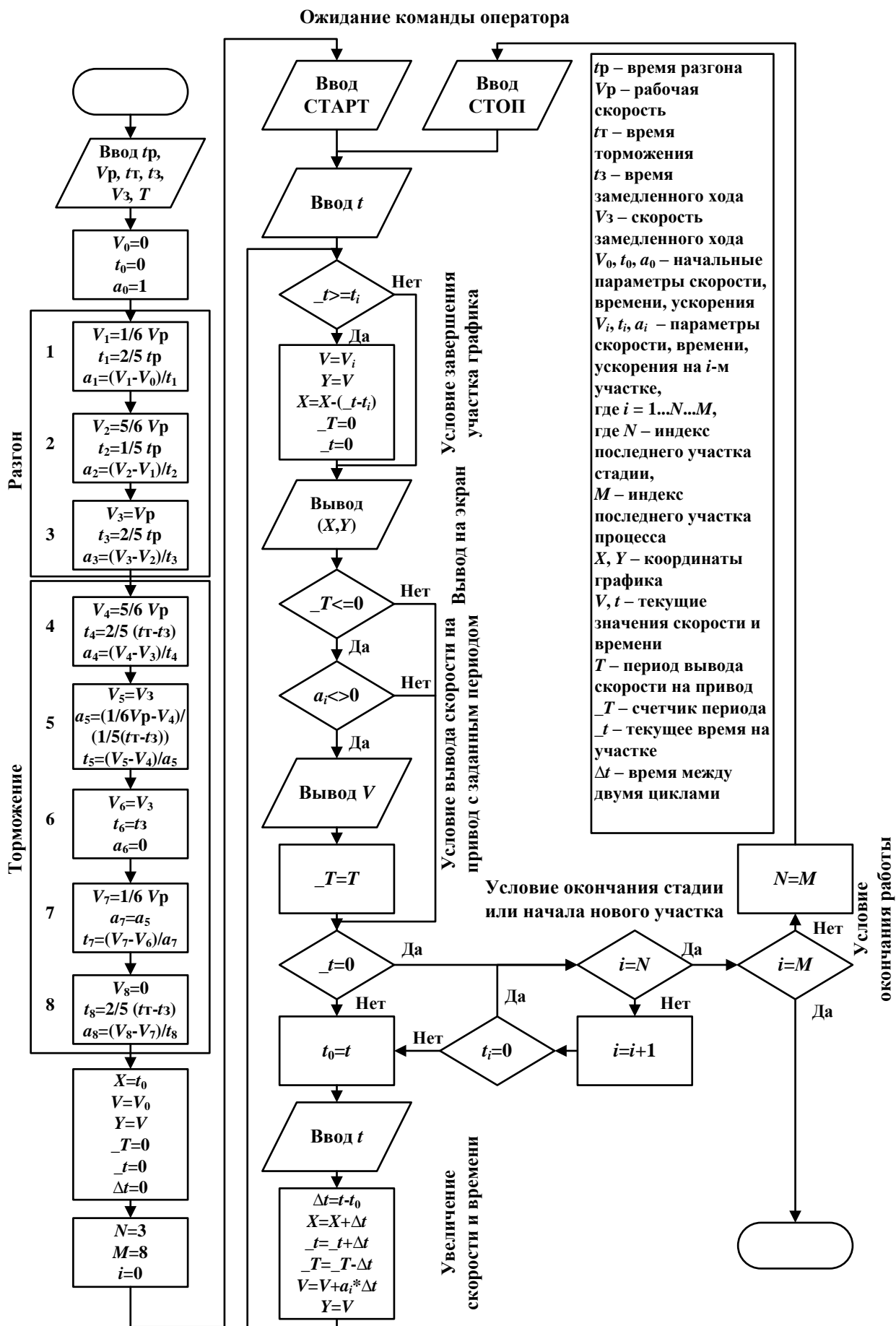


Рис. 4.5. Алгоритм управления разгоном – торможением привода

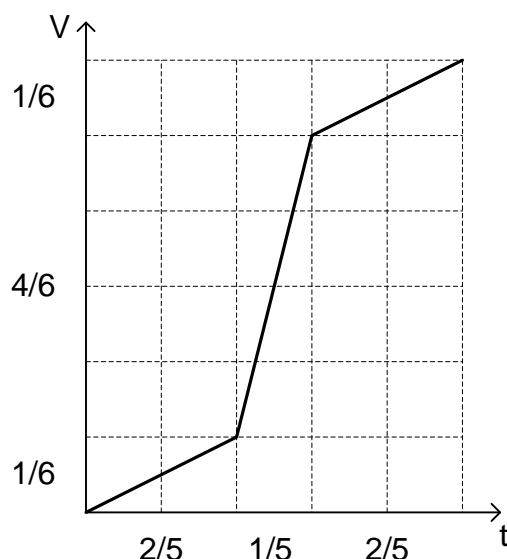


Рис. 4.6. Распределение скорости и времени по долям

Такие соотношения обеспечивают универсальность кривой и позволяют добиваться наиболее плавного разгона и торможения в большинстве практических случаев. Однако в случае торможения во 2-й участок с максимальным замедлением привносится дополнительная стадия замедленного хода, разбивающая этот участок на две части (отсюда 5 частей стадии торможения). Включение ее в процесс торможения обусловливается более существенной ролью последнего в рамках перемещения груза с точки зрения как точности позиционирования, так и нагрузок, при этом возникающих.

При задании параметров оператором значение времени замедленного хода (t_3) включается в общее значение времени торможения (t_T) и не может его превышать (но может быть равным нулю), при этом суммарная доля 2-го и 4-го участка торможения вычисляется как $1/5(t_T - t_3)$, а 1-я и 5-я части — как $2/5(t_T - t_3)$ каждая. Скорость же замедленного хода (V_3), как видно из графика, не может превышать $5/6 V_p$ и быть меньше $1/6 V_p$.

Формулы, приведенные в алгоритме для 8 частей, определяют максимальное/минимальное значение скорости на i -м участке V_i как долю рабочей скорости V_p ; временной промежуток для i -го участка t_i при разгоне — как долю времени t_p и при торможении — как долю времени $(t_T - t_3)$;

ускорение – как соотношение изменения скорости на i -м участке $(V_i - V_{i-1})$ и времени t_i . Все это относится ко всем участкам, кроме 5, 6 и 7. На 5-м участке (участок перед замедленным ходом) вначале минимальная скорость V_5 приравнивается к замедленной скорости V_3 , затем высчитывается ускорение a_5 , определяемое отношением изменения скорости на всем разбитом участке с максимальным ускорением $(1/6V_p - 5/6V_p)$ ко времени $1/5(t_T - t_3)$, и только после этого высчитывается время $t_5 = (V_5 - V_4)/a_5$. Для 6-го участка (замедленный ход) скорость V_6 и время t_6 определяются значениями V_3 и t_3 , а ускорение равно 0. Для 7-го участка (после замедленного хода) минимальная скорость V_7 стандартно определена как $1/6V_p$, ускорение a_7 точно такое же, как на 5-м участке, а время $t_7 = (V_7 - V_6)/a_7$.

Для построения графика изменения скорости на экране и вывода скорости в реальном времени на привод применяется центральная часть алгоритма (рис. 4.5).

Предварительно в конце левой части алгоритма настраиваются (обнуляются) отдельные параметры: координаты X и Y ; начальное значение скорости V ; счетчик периода $_T$; текущее время на отдельном участке $_t$; время между циклами Δt ; также устанавливаются значения количества участков N и M и счетчика участков i .

После этого программа ожидает ввода пользователем команды старта, при поступлении которой фиксируется начальное время t , выдаваемое системным таймером.

Далее по условию $_t \geq t_i$ определяется, завершился текущий участок или нет. Вначале, поскольку V_0 , t_0 и $_t$ равны нулю, значения V , X и Y остаются равными нулю, далее на экран и привод (по условию $_T \leq 0$) выводятся их нулевые значения, задавая начальную точку на экране, сбрасывая контроллер и устанавливая привод в состояние готовности к работе.

После этого по условию $_t = 0$ осуществляется переход на следующий (при запуске 1-й) участок ($i = i+1$). Затем в переменной t_0 запоминается предыдущее значение времени t , считывается настоящее значение t и

определяется $\Delta t = t - t_0$, после чего в соответствии со значением Δt устанавливаются значения скорости V и координаты Y ($V = V + a_i \cdot \Delta t$ и $Y = V$), а также изменяются временные значения координаты X , локального времени участка $_t$ и счетчика периода $_T$ ($X = X + \Delta t$, $_t = _t + \Delta t$, $_T = _T - \Delta t$).

Далее алгоритм возвращается к верхней точке и проверяет, не превысило ли время работы на участке ($_t$) время, отведенное на этот участок (t_i). Внутри условия (при завершении текущего участка) происходит установка значений скорости V и координат X и Y в соответствии с параметрами заверченного участка, а также обнуление локального времени участка $_t$ (свидетельствуя о конце участка и готовности к началу нового) и счетчика периода $_T$ (обеспечивая вывод на привод значения скорости в конце участка вне зависимости от реального завершения периода).

Здесь необходимо отметить, что, несмотря на весьма небольшие промежутки между циклами, реальное время на участке если и незначительно, но превысит время t_i (обеспечить идеальную временную подгонку не представляется возможным). Именно в связи с этим возникает необходимость корректировки скорости V и координат X и Y , устанавливая их в крайнее для заверченного участка положение: $V = V_i$, $Y = V$, $X = X - (_t - t_i)$, где $(_t - t_i)$ – значение превышения по времени. Таким образом, на привод в конце будет выводиться заданная минимальная/максимальная скорость на участке (в том числе максимальная рабочая и ноль), а небольшое временное превышение существенно не повлияет на характер изменения скорости.

Далее по алгоритму на экран выводятся значения X и Y . Стоит обратить внимание на то, что вывод на экран осуществляется в масштабированном виде в зависимости от разрешения экрана, значения рабочей скорости V_p (а при соответствующей настройке и от поддерживаемого ЦАПом диапазона скоростей) и суммарного времени $tp + t_T$ (с добавлением некоторого условного времени работы на скорости V_p), исходя из которых опре-

деляется число условных единиц на пиксель по оси Y и секунд на пиксель по оси X .

Вывод нового значения скорости на привод осуществляется только при обнулении счетчика или его отрицательном значении ($_T \leq 0$), а также при условии действительного наличия изменения скорости на участке ($a_i \neq 0$), что исключает повторную запись в контроллер привода ранее записанного и установленного значения скорости. После вывода на привод скорости счетчик периода заново запускается ($_T = T$).

Далее при соблюдении условия $_t = 0$ идет проверка на возможность начала нового участка, переход на который осуществляется в том случае, если на данной стадии участки еще имеются ($i < N$). В случае же завершения стадии разгона или торможения ($i = N$) уточняется, не является ли этот участок последним во всем процессе ($i = M$), и если является, то алгоритм завершает работу (процесс разгона – торможения привода успешно завершился). Если же это лишь окончание первой стадии (разгона), значение N увеличивается до M (N после этого будет соответствовать номеру последнего участка на стадии торможения) и программа переходит к ожиданию остановки привода оператором.

Если участок не последний на стадии (т. е. $i < N$), то индекс i увеличивается, осуществляя переход на начало нового участка. Однако перед этим осуществляется еще одна проверка: если время на новом участке равно нулю ($t_i = 0$), то тут же осуществляется переход на следующий участок. В нашем случае подобное происходит при равенстве времени на участке замедленного хода нулю ($t_6 = 0$). Таким образом, алгоритм остается цельным и не тратится цикл на обработку нулевого промежутка.

Если переход на новый участок все-таки произведен или старый участок продолжается, то определяется Δt и все значения меняются в соответствии с новым Δt и новым или старым a_i . После чего цикл начинается сначала (с проверки условия завершения участка).

Описанный алгоритм был реализован в программном обеспечении «Система разгона–торможения управляемого привода». Программное обеспечение создавалось в среде программирования Visual Studio 2008 на языке программирования C++. Решение *Drive Accel-Decel Control (DADC)* включает в себя три формы *Windows Forms* (главное окно, окно справки и титульное окно программы), описанные в файлах *DADC.h*, *DADC_Help.h*, *DADC_Front.h*. В дополнение к ним в файле *DADC_Plot.h* был создан класс обработки заданных участков с выводом на экран в виде графика и функцией вывода на привод (средняя часть алгоритма). Задание же и расчет значений параметров V , t и a запрограммированы в коде файла *DADC.h* главной формы.

Вначале в функции *InitializePlot* происходит инициализация нового графика *pPlot*, затем при сохранении посредством функции *bSave_Click* в график записывается алгоритм работы в соответствии с заданными пользователем значениями t_p , V_p , t_T , t_z , V_z с разбиением на стадии и участки (*States* и *Segments*) и указанием значений V , t и a для каждого участка.

Здесь же описывается визуальное оформление графика, а до начала сохранения проверяется корректность введенных данных и выдается сообщение об ошибке при ее наличии.

После успешного сохранения, создания в памяти графика изменения скорости и прорисовки его на экране (без вывода на привод) программа ожидает нажатия пользователем кнопки *START* (или повторного изменения параметров и сохранения).

Нажатие кнопки *START* через функцию *bStart_Click* или в дальнейшем кнопки *STOP* через функцию *bStop_Click* запускает управляющую часть алгоритма с начала (разгон) либо с продолжения (торможение): используется функция *Start* класса *Plot* (файл *DADC_Plot.h*), которая в свою очередь последовательно вызывает функцию *Update*, проходя через все заданные участки до момента окончания стадии и вывода через промежуток

Δt значения скорости на экран функцией *Draw* и на привод с периодом T функцией *Output*.

Код программы с комментариями к конкретным функциям и объектам приводится в приложении.

Система предназначена для использования в программируемых контроллерах, управляющих регулируемыми приводами рабочих машин различного назначения. Подключение системы к приводу производится через ЦАП на подготовительной стадии и после запуска программы осуществляется автоматически. При отсутствии ЦАП и привода система работает в тестовом режиме.

Система представляет собой программу, содержащую 3 файла: инструкцию по эксплуатации (*Инструкция.rtf*), файл последних сохраненных параметров (*DADC.dat*) и исполняемый файл (*Drive Accel-Decel Control.exe*). Исполняемый файл запускается в ОС Windows. При запуске появляется окно с информацией о названии системы и ее разработчиках с двумя кнопками: *Справка* (выводит окно с инструкцией) и *Далее >>>* (запускает саму программу).

После нажатия при помощи мыши или клавиши Enter на кнопку *Далее >>>* на весь экран выводится главное окно системы. В верхней его части располагаются управляющие элементы системы, под ними прорисовывается график изменения скорости при разгоне и торможении, значения которой выводятся на ЦАП и управляемый привод (рис. 4.7). На участке разгона график реализует возрастающую кубическую параболу, аппроксимированную тремя отрезками прямых, на участке торможения – убывающую кубическую параболу, аппроксимированную аналогично, но с дополнительным горизонтальным участком, соответствующим замедленной скорости.

В левом верхнем углу окна расположены кнопки управления приводом: зеленая кнопка *START*, запускающая привод, и красная кнопка *STOP*,

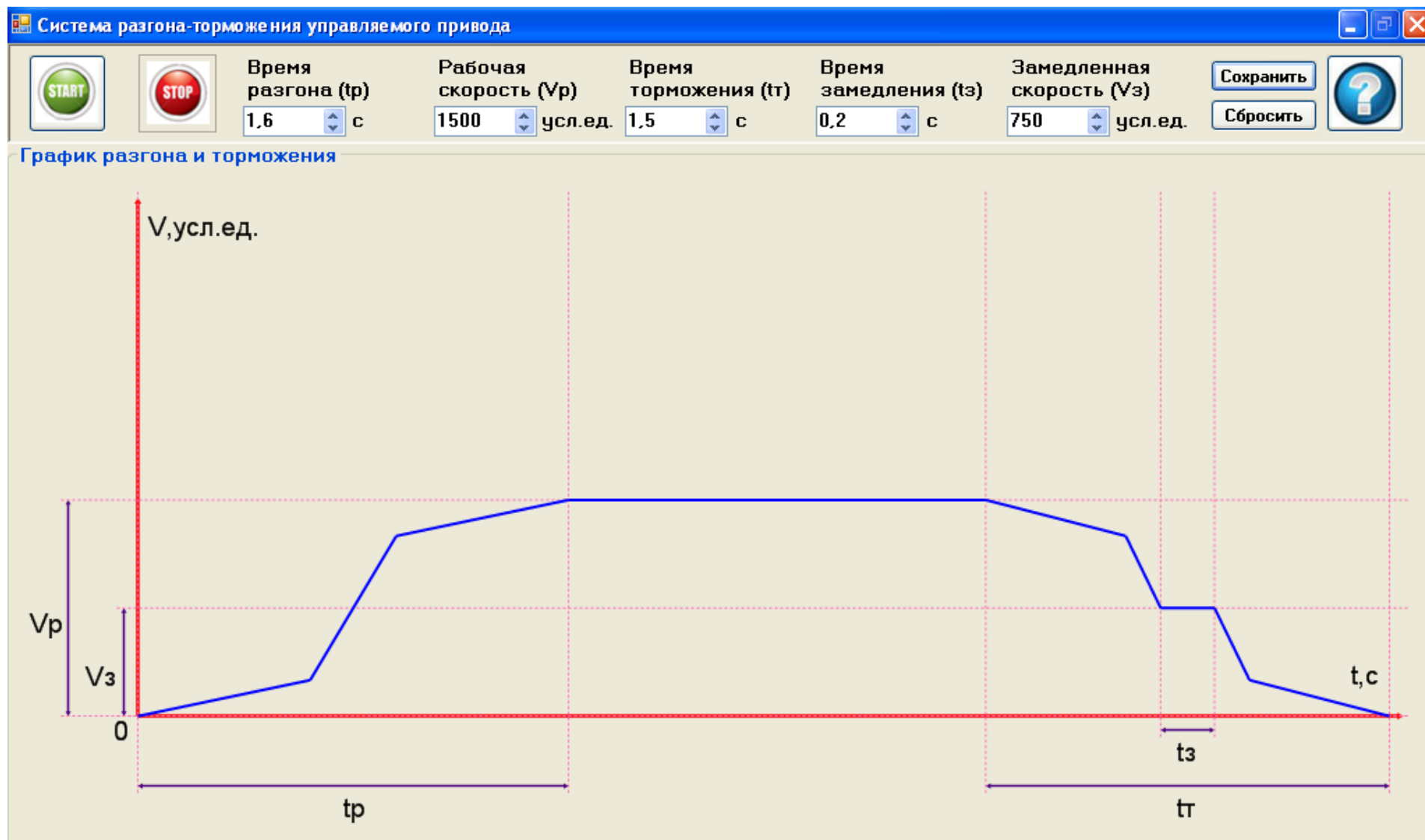


Рис. 4.7. Главное окно системы

его останавливающая. Вверху в середине окна находятся *поля отдельных параметров* разгона и торможения, значения которых устанавливаются оператором, справа вверху – кнопка *Сохранить*, при нажатии которой рассчитываются и устанавливаются значения времени и скорости для отдельных участков графика изменения скорости перед нажатием кнопки *START*, кнопка *Сбросить*, восстанавливающая предыдущие значения параметров, и кнопка *Справки* (знак вопроса на синем фоне), выводящая инструкцию.

Перед запуском привода оператору необходимо установить параметры работы. Их начальные (при первом запуске программы) значения равны нулю. После нажатия на кнопку *Сохранить* последние измененные значения записываются в файл *DADC.dat* и при следующем запуске программы загружаются из него в поля ввода автоматически. Вручную изменять этот файл не рекомендуется. При наличии проблем его достаточно удалить – при сохранении программа заново его восстановит.

Система позволяет ввести 5 параметров, управляющих приводом: *время разгона (t_p)*, *рабочую скорость (V_p)*, *время торможения (t_T)*, *время замедления (t_z)*, *замедленную скорость (V_z)*.

Ввод значений в поле может осуществляться пользователем как непосредственно с клавиатуры, так и при помощи кнопок *вверх-вниз*, расположенных справа от поля, используя для этого либо мышь, либо стрелки вверх-вниз на клавиатуре. Кнопки *вверх-вниз* увеличивают или уменьшают числовое значение на единицу последнего разряда целого числа или дроби (для введения дробного значения нужно ввести с клавиатуры в окно запятую или точку в зависимости от региональных настроек системы).

При запуске программы первоначальный фокус получает поле *Время разгона*. Переход на следующее поле осуществляется при помощи мыши или клавиши *Tab*, при нажатии которой фокус последовательно слева направо по кругу перемещается во все управляющие элементы программы. Введенные в поля значения остаются в них, но сохраняются только при нажатии на кнопку *Сохранить* щелчком мыши или клавишей *Enter*.

Поля времени подразумевают ввод значений времен в реальных единицах времени – секундах, *поля скорости* – в условных единицах скорости, которые зависят от дискретности ЦАП и характеристик привода. В программе предусмотрены ограничения на ввод значений отдельных параметров. Если ограничения нарушаются, то после нажатия на кнопку *Сохранить* на экран выводится мигающее сообщение об ошибке, указывается поле, в котором она произошла, и неверно введенное значение, которое необходимо изменить. После этого оператор может ввести другое значение и заново проверить его корректность через сохранение либо нажать кнопку *Сбросить*, которая возвращает систему к предыдущим корректным значениям. Эту кнопку также можно использовать и после успешного сохранения при неудовлетворительной форме графика или даже после прогона привода в рабочем режиме, если работа по установленным параметрам пользователя не устраивает.

Так, значения времен разгона и торможения, а также рабочей скорости должны быть больше нуля. А точнее, быть больше периода вывода значения скорости на привод. При этом изменение периода возможно только при непосредственном редактировании файла *DADC.dat* (последнее число в нем), и в общем случае установленного по умолчанию значения 0,0001 сек достаточно для нормальной работы контроллера и привода. Временной промежуток работы на замедленной скорости (поле t_z) может быть как равным нулю (дополнительная стадия отсутствует), так и любым другим положительным числом, не превышающим общее время торможения t_T , значение которого включает в себя t_z . Значение же скорости замедленного хода V_z при $t_z = 0$ не имеет значения и может быть любым. При использовании режима замедленной скорости значение V_z должно быть установлено в пределах между $1/6$ и $5/6$ рабочей скорости (точное значение указывается при ошибке ввода V_z).

По завершении ввода всех или только отдельных параметров перед запуском двигателя эти параметры необходимо сохранить (кнопка *Сохра-*

нить). При отсутствии ошибок (о которых сигнализируется пользователю в области графика) программа произведет расчет значений времени, скорости и ускорения для всех стадий, после чего на графике происходит полная прорисовка изменения скорости во времени в соответствии с установленными параметрами как на стадии разгона, так и на стадии торможения (но не в реальном времени и без вывода скорости на привод).

На полученном в результате сохранения графике прорисованы две оси: горизонтальная – время t, c и вертикальная – скорость $V, \text{усл.ед.}$. Слева от вертикальной оси и внизу под горизонтальной указаны выносные размеры, соответствующие тем значениям, которые вводятся в поля, а штрихпунктирная сетка в середине графика дает полное представление об этих параметрах и стадиях будущего процесса.

Левая часть графика будет прорисовываться в режиме реального времени после нажатия клавиши *START*. Запуск привода этой кнопкой может быть осуществлен оператором непосредственно после успешного сохранения параметров. До сохранения параметров программа не позволит нажать эту кнопку. То же и до завершения процессов разгона–торможения при уже запущенном приводе: кнопку *STOP* возможно нажать только по завершении процесса разгона. Визуально готовность кнопки к нажатию отображается ее полной прорисовкой, состояние нажатия – «утопленностью» кнопки, а состояние неготовности к нажатию – «выпуклостью» (в последних двух случаях нажатие происходит вхолостую, т. е. ни к чему не приводит). После завершения процесса разгона–торможения кнопка *START* готова к повторному использованию оператором, который может продолжать работать на тех настройках, которые были установлены. До момента завершения процессов разгона и торможения в программе отключается также возможность переключения на другие окна и свертывания окна при помощи мыши.

Нажатие кнопки *START* запускает в реальном времени процесс разгона двигателя точно по левой части графика изменения скорости во вре-

мени, отображенной на экране после сохранения параметров. По завершении разгона на экран выводится центральная часть графика, соответствующая использованию механизма на рабочей скорости и в рабочем режиме в течение некоторого формального промежутка времени (на экране прорисовывается мгновенно). Прорисовка правой части графика происходит после начала торможения (нажатия кнопки *STOP*), производимого оператором после завершения работы. После остановки оператором могут быть выполнены дальнейшие действия: повторения процесса работы (разгон–работа–торможение), изменение параметров с их сохранением, сброс параметров к предыдущим значениям с их сохранением, выход из программы по завершении работы (параметры в файл уже сохранены).

Выход из программы осуществляется стандартным путем при нажатии на «крестик» в правом верхнем углу окна либо сочетанием *Alt-F4*.

В общем случае задача оператора на подготовительной стадии (или даже в процессе работы) сводится к подбору экспериментальным путем таких параметров, которые бы обеспечивали наиболее плавные разгон и торможение на данном двигателе при данных эксплуатационных характеристиках крана и условиях работы с возможным варьированием значений в зависимости от массы поднимаемого груза.

5. ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ СРЕДСТВ УСТРАНЕНИЯ КОЛЕБАНИЙ ГРУЗОВ И СНИЖЕНИЯ АВАРИЙНОСТИ И ТРАВМАТИЗМА ПРИ ЭКСПЛУАТАЦИИ КРАНОВ

Поскольку грузоподъемный кран непосредственно материальную продукцию не выпускает, использование всякого крана по его служебному назначению можно рассматривать как продажу специфических услуг – услуг по производству подъемно-транспортных работ. Эти работы в случае их выполнения по договорам подряда с некоторыми заказчиками за период

времени T , мес, с начала эксплуатации крана приносят его владельцу прибыль

$$\Pi = \Pi \cdot T - P,$$

где Π – цена указанных работ, руб/мес; P – расходы и потери, руб, связанные с эксплуатацией крана, за время T . Определим величину P как

$$P = A + \Xi + \Pi_{\Pi} + \Pi_T,$$

где A – амортизационные отчисления; Ξ – собственно эксплуатационные затраты; Π_{Π} – потери из-за простоев крана; Π_T – потери по причине производственного травматизма персонала, обслуживающего кран.

Согласно Положению по бухгалтерскому учету «Учет основных средств» ПБУ 6/01, утвержденному Приказом Минфина России от 30.03.2001 № 26н (пп. 19, 20), A рассчитывается по первоначальной стоимости крана C и норме амортизации H , исчисленной исходя из срока полезного использования крана T_{Π} , который устанавливается предприятием-владельцем крана при принятии его на баланс в соответствии с Постановлением Правительства РФ от 01.01.2002 №1 «О классификации основных средств, включаемых в амортизационные группы». Отсюда следует, что в течение времени T , не превышающего T_{Π} ,

$$A = \frac{C \cdot H \cdot T}{100},$$

где C – величина, включающая в себя цену крана, затраты на его транспортирование к месту монтажа и монтаж, руб; H – месячная норма амортизационных отчислений, выраженная в процентах. Если принять во внимание, что по истечении срока T_{Π} амортизационные отчисления прекращаются, то в общем случае (для произвольных T)

$$A = \frac{C \cdot H}{100} \cdot K_A,$$

где K_A – пороговая функция

$$K(T, T_{\Pi}) = \begin{cases} T & \text{при } T < T_{\Pi} \\ T_{\Pi} & \text{при } T \geq T_{\Pi} \end{cases}.$$

Собственно эксплуатационные затраты \mathcal{E} складываются из затрат на текущие ремонты и небольшие модернизации крана (крупные модернизации выполняются во время капитальных ремонтов и производятся за счет амортизационных отчислений), из оплаты персонала, обслуживающего кран, затрат на электроэнергию и вспомогательные материалы (тросы, смазочные материалы и др.) и прочее и приближенно могут быть определены как

$$\mathcal{E} = \mathcal{Z}_1 \cdot T,$$

где \mathcal{Z}_1 – средние собственно эксплуатационные затраты в месяц, руб.

Потери из-за простоев крана определить сложнее, но с точностью, достаточной для практики, их можно вычислить как

$$\Pi_{\Pi} = \mathcal{C} \cdot \frac{T_{\Pi\mathcal{K}}}{\Phi_{\mathcal{K}}} \cdot T + \mathcal{Z}_2 \cdot \frac{T_{\Pi\mathcal{K}\mathcal{A}}}{\Phi_{\mathcal{K}}} \cdot T,$$

где $T_{\Pi\mathcal{K}}$ – среднегодовое время простоев крана по организационным (по вине владельца) и техническим (крупные неисправности и аварии) причинам; $T_{\Pi\mathcal{K}\mathcal{A}}$ – среднегодовое время простоев крана только по техническим причинам (то и другое в часах); \mathcal{Z}_2 – среднемесячные затраты на устранение крупных неисправностей и последствий аварий, руб; $\Phi_{\mathcal{K}}$ – годовой фонд рабочего времени крана, ч.

Потери из-за травматизма персонала, обслуживающего кран, согласно работе А. А. Вершинина, Я. Л. Либермана, В. П. Ситникова и др. [30], можно вычислить аналогично, как

$$\Pi_{\mathcal{T}} = \mathcal{Z}_3 \cdot \frac{T_{\Pi\mathcal{T}}}{\Phi_{\mathcal{P}}} \cdot T,$$

где $T_{\Pi\mathcal{T}}$ – среднегодовые потери рабочего времени указанного персонала по причине травматизма; $\Phi_{\mathcal{P}}$ – годовой фонд рабочего времени персонала (то и другое в часах); \mathcal{Z}_3 – средняя оплата месяца нетрудоспособности в связи с производственным травматизмом персонала, обслуживающего кран, осуществляемая Фондом социального страхования РФ из страховых

вносов предприятия наряду с возвратом предприятию средств для реализации предупредительных мер по сокращению травматизма и профзаболеваний (Приказы Минздравсоцразвития РФ от 30.01.2008 № 43н и 06.08.2009 № 573н).

Таким образом, Р и П равны соответственно

$$P = \left(Z_1 + Ц \cdot \frac{T_{ПК}}{\Phi_K} + Z_2 \cdot \frac{T_{ПКА}}{\Phi_K} + Z_3 \cdot \frac{T_{ПТ}}{\Phi_P} \right) \cdot T + \frac{C \cdot H}{100} \cdot K_A,$$

$$П = Ц \cdot T \cdot \left(1 - \frac{T_{ПК}}{\Phi_K} \right) - T \cdot \left(Z_1 + Z_2 \cdot \frac{T_{ПКА}}{\Phi_K} + Z_3 \cdot \frac{T_{ПТ}}{\Phi_P} \right) - \frac{C \cdot H}{100} \cdot K_A.$$

Зная Р и П, нетрудно найти прирост прибыли, сокращение расходов, связанных с эксплуатацией, и другие показатели, характеризующие экономическую эффективность применения тех или иных технических средств повышения производительности, надежности, безопасности и иных качеств грузоподъемных кранов. Так, прирост прибыли будет равен

$$\Delta П = П'' - П',$$

где П' – прибыль до внедрения технических средств; П'' – прибыль после внедрения, отличающаяся от П' за счет изменения Z_1 , $T_{ПК}$, $T_{ПКА}$, $T_{ПТ}$ или $T = T_{СЛ}$, где $T_{СЛ}$ – срок службы крана, изменившийся за счет увеличения его остаточного ресурса. Полагая, что Р применительно к грузоподъемным машинам имеет смысл себестоимости продукции, можно рассчитать условно-годовую экономию от использования тех или иных технических средств или организационных мероприятий как

$$\mathcal{E}_{\text{УГ}} = \frac{P' - P''}{T} \cdot 12,$$

срок окупаемости капиталовложений М, лет, потребовавшихся на внедрение некоторого новшества, как

$$T_{\text{ОК}} = \frac{M \cdot T}{(P' - P'') \cdot 12},$$

где Р' и Р'' – величины, соответствующие П' и П'' при равных Т, а также

вычислить относительное снижение расходов и потерь как

$$\Delta P_{\text{отн}} = \left(1 - \frac{P''}{P'}\right) \cdot 100 (\%).$$

Нетрудно рассчитать и коэффициент рентабельности затрат P и его изменение как

$$\Delta K_p = K_p'' - K_p' = \left(\frac{\Pi''}{P''} - \frac{\Pi'}{P'}\right) \cdot 100 (\%).$$

Оценим теперь по приведенной методике экономическую целесообразность применения устройств устранения колебаний, описанных в гл. 4.

Будем считать, что все они приблизительно равноценны и M для любого из них составляет 5 000 руб. Примем во внимание, что в период простоев кранов начисление амортизации не прерывается, если простой длится не более трех месяцев (в противном случае кран должен быть поставлен на консервацию), причем расходы на транспортировку крана и монтаж при расчете C обычно полагают равными соответственно 20 и 10 % от цены крана [31].

Предположим далее, что имеется башенный кран грузоподъемностью 16 т и первоначальной стоимостью 1 млн руб. Полагая, что затраты на транспортировку крана к месту работ составляют примерно 200 тыс. руб. и его монтаж на рабочей площадке примерно 100 тыс. руб., примем величину C в формуле расчета амортизационных отчислений равной 1 300 000 руб.

По Постановлению Правительства РФ от 01.01.2002 №1 башенные краны общего назначения относятся к подразделу «Машины и оборудование – Оборудование подъемно-транспортное подвижное» 5-й группы, описывающей «имущество со сроком полезного использования свыше 7 лет до 10 лет включительно». Приняв срок полезного использования крана равным 10 годам ($T_{\text{ПИ}} = 10 \cdot 12 \text{ мес} = 120 \text{ мес}$), получим годовую норму амортизационных отчислений в размере 10 %, а месячную норму $H = 10 / 12 \approx 0,83 \%$.

Для определения средних эксплуатационных затрат в месяц Z_1 зададим их составляющие: затраты на текущие ремонты и небольшие модернизации крана в размере 7 500 руб/мес; расходы на вспомогательные материалы – 5 000 руб/мес; затраты на оплату персонала для бригады из 3 человек на 1 кран и среднемесячной зарплатой 20 тыс. руб – 60 000 руб; расходы на электроэнергию для крана с суммарной установленной мощностью электродвигателей 150 кВт при стоимости 0,5 руб за 1 кВт и годовом фонде рабочего времени крана Φ_K , равном 1 800 ч, – $150 \cdot 0,5 \cdot 1800/12 = 11250$ руб/мес.

Таким образом, $Z_1 = 7500 + 5000 + 60000 + 11250 = 83750$ руб.

Исходя из статистики аварийности и травматизма, приведенной в гл. 1, примем вероятность $T_{ПК}/\Phi_K$ простоя крана по организационным и техническим причинам равной 0,15, а вероятность $T_{ПКА}/\Phi_K$ простоя по техническим причинам – 0,10. Среднегодовые потери рабочего времени персонала по причине травматизма $T_{ПТ}$ можно вычислить также по материалам гл. 1 из расчета 6 несчастных случаев в год (для бригады из 3 человек) и средней продолжительности одного больничного листа – 2 нед., что при 40-часовой рабочей неделе будет равняться $6 \cdot 2 \cdot 40 = 480$ ч/г. Тогда при номинальном фонде рабочего времени 250 рабочих дней в году или 2000 часов в году на человека [32] и, соответственно, общем фонде рабочего времени Φ_P , равном $3 \cdot 2000 = 6000$ часов, можно рассчитать вероятность потерь по причине травматизма как $T_{ПТ}/\Phi_P = 480/6000 = 0,08$.

Зададимся далее величинами Π , Z_2 и Z_3 . Величину Z_3 приравняем к указанной выше среднемесячной зарплате персонала (60 000 руб.), поскольку больничные листы при производственном травматизме оплачиваются в размере 100 % среднего заработка работника. Значение Z_2 , исходя из среднегодовых затрат на устранение последствий аварий, составляющих около 25 % от стоимости крана, примем равным 20 000 руб/мес. Наконец,

исходя из соображений прибыльности, примем величину Ц равной 200 000 руб/мес.

Определим теперь расходы и потери, связанные с эксплуатацией крана без применения средств устранения колебаний. Если считать, что срок службы крана в данном случае равен $T_{\text{СЛ}} = T'_{\text{СЛ}} = 12 \text{ лет} = 144 \text{ мес}$, то при выбранных выше исходных данных Р равно

$$P' = \left(83750 + 200000 \cdot \frac{270}{1800} + 20000 \cdot \frac{180}{1800} + 60000 \cdot \frac{480}{6000} \right) \cdot 144 + \frac{1300000 \cdot 0,83}{100} \cdot 120 \\ = 18659200 \text{ руб.}$$

Соответственно, прибыль П, полученная владельцем крана за весь срок службы последнего, будет равна

$$П' = 200000 \cdot 144 - 18659200 = 10140800 \text{ руб.}$$

Из исследований технической эффективности устройств устранения колебаний, описанных в работе [1], известно, что они могут увеличить срок службы крана $T_{\text{СЛ}}$ приблизительно на 15% (в рассматриваемом примере до $T''_{\text{СЛ}} = 13,8 \text{ лет}$ или 165,6 мес), величину $T_{\text{ПТ}}$ позволяют снизить также на 15% (до $T''_{\text{ПТ}} = 408 \text{ ч/г}$), а величину $T_{\text{ПКА}}$ – на 10% ($T''_{\text{ПКА}} = 162 \text{ ч/г}$). Отсюда следует, что вероятность простоев крана по техническим причинам в результате применения средств, предложенных в гл.4, может уменьшиться до величины $T''_{\text{ПКА}}/\Phi_K = 0,09$, вероятность простоев по организационным и техническим причинам (при неизменной вероятности простоев по оргпричинам, равной примерно 0,05) – до величины $T''_{\text{ПКА}}/\Phi_K + 0,05 = 0,14$, вероятность $T_{\text{ПТ}}/\Phi_P$ – до величины $T''_{\text{ПТ}}/\Phi_P = 0,07$. Подставляя в формулы для Р и П числа, соответствующие этим вероятностям, и полагая, что затраты на столь малую модернизацию компенсируются уменьшением затрат на текущие ремонты крана, получим за весь новый срок службы крана

$$P'' = \left(83750 + 200000 \cdot \frac{252}{1800} + 20000 \cdot \frac{162}{1800} + 60000 \cdot \frac{408}{6000} \right) \cdot 165,6 + \frac{1300000 \cdot 0,83}{100} \cdot 120$$

$$= 20779528 \text{ руб.},$$

$$\Pi'' = 200000 \cdot 165,6 - 20779528 = 12340472 \text{ руб.}$$

Сравнивая P'' и P' (рис. 5.1), нельзя не заметить, что установка на кран устройств устранения колебаний суммарные расходы и потери увеличила. Однако это произошло за счет увеличения срока службы крана. Но увеличение срока, уменьшение уровня травматизма и времени простоев крана привело и к получению владельцем крана финансовой выгоды в виде прироста прибыли (рис. 5.2)

$$\Delta\Pi = 12340472 - 10140800 = 2199672 \text{ руб.}$$

Отсюда видно, что экономическая эффективность устройств, предназначенных для устранения колебаний, оказывается весьма существенной. Для подтверждения этого можно вычислить и другие показатели, характеризующие экономическую эффективность таких устройств, – $\mathcal{E}_{\text{УГ}}$, $T_{\text{ОК}}$, $\Delta P_{\text{ОТН}}$ и ΔK_p . Это нетрудно сделать, если задаться некоторым расчетным временем эксплуатации крана T , одинаковым для сравниваемых случаев. При $T = 5 \text{ лет} = 60 \text{ мес}$, например, будем иметь до модернизации крана

$$P' = \left(83750 + 200000 \cdot \frac{270}{1800} + 20000 \cdot \frac{180}{1800} + 60000 \cdot \frac{480}{6000} \right) \cdot 60 + \frac{1300000 \cdot 0,83}{100} \cdot 60$$

$$= 7883000 \text{ руб.},$$

$$\Pi' = 200000 \cdot 60 - 7883000 = 4117000 \text{ руб.}$$

После модернизации получается

$$P'' = \left(83750 + 200000 \cdot \frac{252}{1800} + 20000 \cdot \frac{162}{1800} + 60000 \cdot \frac{408}{6000} \right) \cdot 60 + \frac{1300000 \cdot 0,83}{100} \cdot 60$$

$$= 7707800 \text{ руб.},$$

$$\Pi'' = 200000 \cdot 60 - 7707800 = 4292200 \text{ руб.}$$

В результате

$$\Delta\Pi = 4292200 - 4117000 = 175200 \text{ руб.},$$

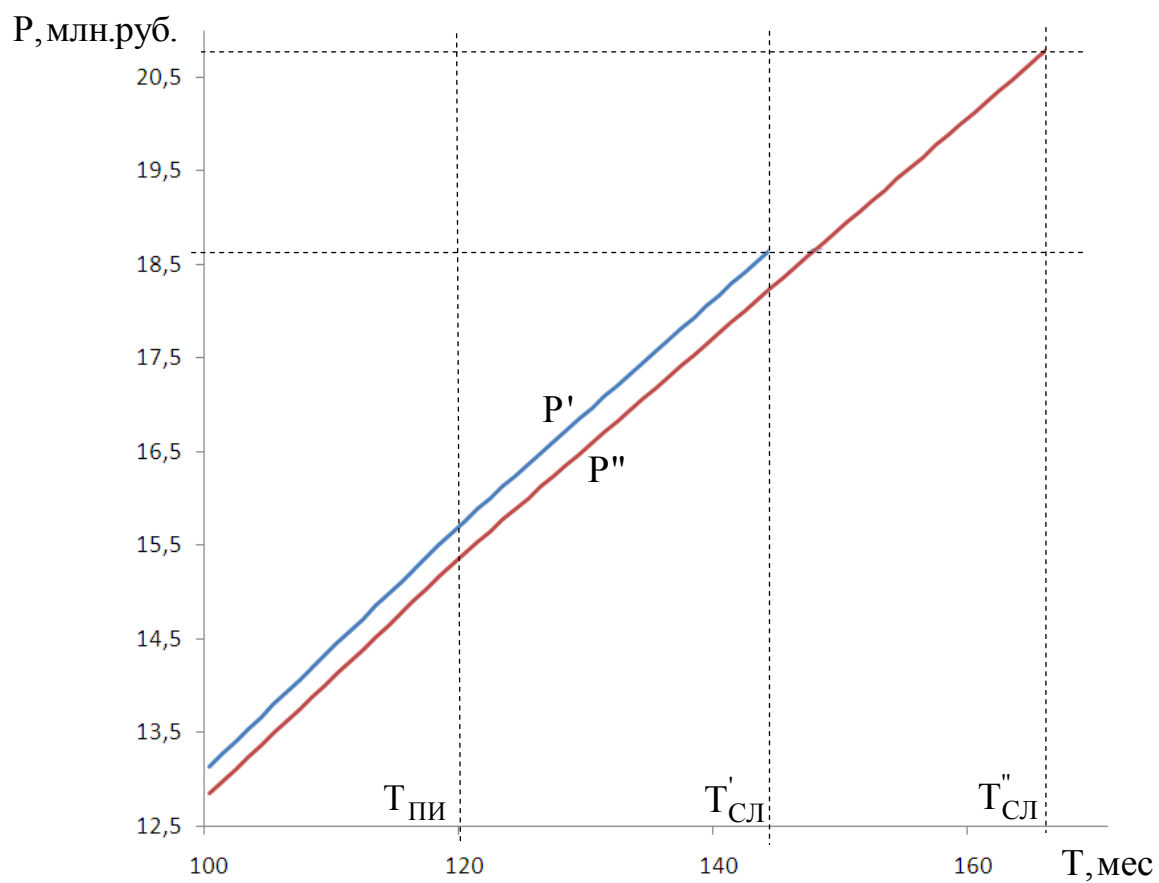


Рис. 5.1. Графики изменения затрат во времени

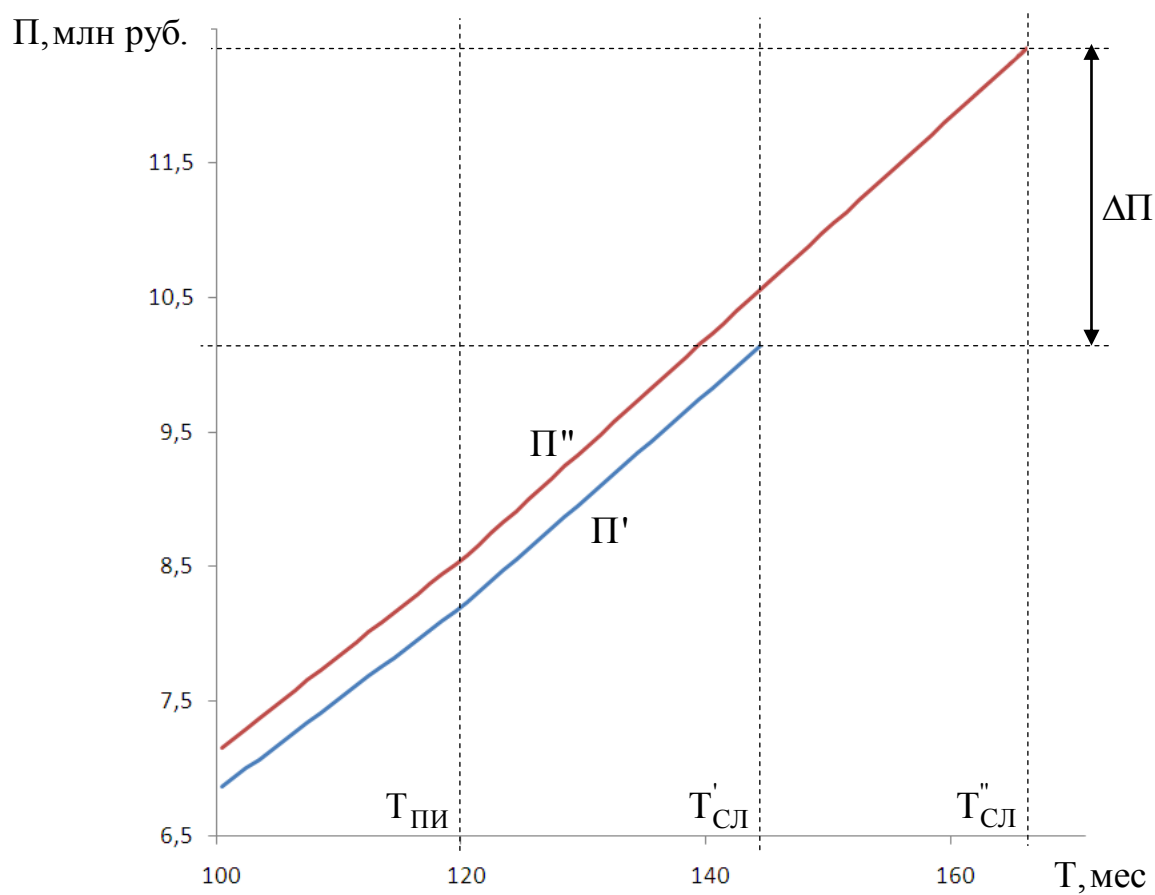


Рис. 5.2. Графики изменения прибыли во времени

$$\mathcal{E}_{\text{yT}} = \frac{7883000 - 7707800}{60} \cdot 12 = 35040 \text{ руб},$$

$$T_{\text{OK}} = \frac{5000 \cdot 60}{(7883000 - 7707800) \cdot 12} \approx 0,14 \text{ г} \approx 1,7 \text{ мес}, ,$$

$$\Delta P_{\text{OTH}} = \left(1 - \frac{7707800}{7883000} \right) \cdot 100 \approx 2,2\%,$$

$$\Delta K_p = \left(\frac{4292200}{7707800} - \frac{4117000}{7883000} \right) \cdot 100 \approx (0,557 - 0,522) \cdot 100 \approx 3,5\%.$$

Полученные значения свидетельствуют об эффективности модернизации не только в расчете на весь период службы крана, но и в пятилетний срок. Разумеется, при других исходных данных и результаты вычислений будут другими. Тем не менее характер их не изменится.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Баранов, В. А. Повышение промышленной безопасности грузоподъемных кранов: дис. ... канд. техн. наук. – СПб.: МАНЭБ, 2009.
2. Кабаков, А. М. Защита машиниста крана от вибрации / А. М. Кабаков, А. И. Пабат, А. Н. Орлов // Безопасность труда в промышленности. 1997. № 5.
3. Либерман, Я. Л. Нетрадиционные методы ликвидации колебаний в рабочих машинах / Я. Л. Либерман, Л. Н. Горбунова, В. А. Баранов // Перспективные процессы и технологии в машиностроении: мат. конф. в рамках IV-й Евро-Азиатской промышленной выставки. Екатеринбург, 2008.
4. А.с. 518589 СССР, М Кл. F16, F15/02. Динамический виброгаситель / Я. Л. Либерман; заявитель Уральский политехн. ин-т – № 2068007/28; заявл. 09.10.74; опубл. 25.06.76. Бюл. № 23.
5. Пат. 87482 Российская Федерация, МПК F16, F15/00. Динамический виброгаситель / Я. Л. Либерман, Н. А. Меньшенина; заявитель и патентообладатель Уральский гос. техн. ун-т – № 2009101612/22; заявл. 19.01.2009; опубл. 10.10.2009. Бюл. № 28.
6. Пат. 87484 Российская Федерация, МПК F16, F15/00. Активный виброгаситель / Я. Л. Либерман; заявитель и патентообладатель Уральский гос. техн. ун-т – № 2009101627/22; заявл. 19.01.2009; опубл. 10.10.2009. Бюл. № 28.
7. Пат. 87483 Российская Федерация, МПК F16, F15/00. Адаптивный виброгаситель / Я. Л. Либерман; заявитель и патентообладатель Уральский гос. техн. ун-т – № 2009101613/22; заявл. 19.01.2009; опубл. 10.10.2009. Бюл. № 28.
8. Чупраков, Ю.И. Гидропривод и средства гидроавтоматики / Ю. И. Чупраков. М.: Машиностроение, 1979.

9. Пат. 87481 Российская Федерация, МПК F16, F15/00. Виброгаситель / Я. Л. Либерман; заявитель и патентообладатель Уральский гос. техн. ун-т – № 2009101659/22; заявл. 19.01.2009; опубл. 10.10.2009. Бюл. № 28.
10. Пат. 88086 Российская Федерация, МПК F16, F15/00. Виброгасящее устройство / Я. Л. Либерман, Н. А. Меньшенина; заявитель и патентообладатель Уральский гос. техн. ун-т – № 2009101661/22; заявл. 19.01.2009; опубл. 27.10.2009. Бюл. № 30.
11. Бешелев, С. Д. Математико-статистические методы экспертных оценок / С. Д. Бешелев, Ф. Г. Гуревич. М.: Статистика, 1974.
12. А.с. 231766 СССР, М Кл. 35а, 21. Способ управления предохранительным торможением шахтных подъемных машин / Е. С. Траубе, Н. В. Каледин; заявители авторы – № 1183355/22-3; заявл. 04.09.67; опубл. 28.11.68. Бюл. № 36.
13. А.с. 716960 СССР, М Кл. В66, В1/24. Способ торможения подъемных машин / И. С. Найдено; заявитель Ин-т горн. мех. и технич. киберн. – № 1677565/29-03; заявл. 06.07.71; опубл. 25.02.80. Бюл. № 7.
14. Пат. 68488 Российская Федерация, МПК В66D 5/26. Устройство ступенчатого торможения подъемно-транспортных машин / Я. Л. Либерман, Л. Н. Горбунова, В. А. Баранов; заявитель и патентообладатель Уральский гос. техн. ун-т – № 2007120316/22; заявл. 30.05.2007; опубл. 27.11.2007. Бюл. № 33.
15. Пат. 2342309 Российская Федерация, МПК В66В 1/24. Способ торможения подъемных машин / Я. Л. Либерман, Л. Н. Горбунова, В. А. Баранов; заявитель и патентообладатель Уральский гос. техн. ун-т – № 2007118613/11; заявл. 18.05.2007; опубл. 27.12.2008. Бюл. № 36.
16. Пат. 68489 Российская Федерация, МПК В66D 5/26. Устройство ступенчатого торможения грузоподъемных машин / Я. Л. Либерман, Л. Н. Горбунова, В. А. Баранов; заявитель и патентообладатель Уральский

- гос. техн. ун-т – № 2007122120/22; заявл. 13.06.2007; опубл. 27.11.2007. Бюл. № 33.
17. Пат. 69055 Российская Федерация, МПК В66D 5/26. Устройство торможения грузоподъемной машины / Я. Л. Либерман, Л. Н. Горбунова, В. А. Баранов; заявитель и патентообладатель Уральский гос. техн. ун-т – № 2007124461/22; заявл. 28.06.2007; опубл. 10.12.2007. Бюл. № 34.
18. Пат. 69054 Российская Федерация, МПК В66D 5/00. Устройство торможения подъемно-транспортной машины / Я. Л. Либерман, Л. Н. Горбунова, В. А. Баранов; заявитель и патентообладатель Уральский гос. техн. ун-т – № 2007126211/22; заявл. 09.07.2007; опубл. 10.12.2007. Бюл. № 34.
19. Пат. 69855 Российская Федерация, МПК В55D 5/00. Устройство торможения подъемной машины / Я. Л. Либерман, Л. Н. Горбунова, В. А. Баранов; заявитель и патентообладатель Уральский гос. техн. ун-т – № 2007124469/22; заявл. 28.06.2007; опубл. 10.01.2008. Бюл. № 1.
20. Пат. 87419 Российская Федерация, МПК В55D 5/00. Устройство двухступенчатого торможения подъемно-транспортной машины / Я. Л. Либерман, Ю. О. Лысый, Л. Н. Горбунова, В. А. Баранов; заявитель и патентообладатель Уральский гос. техн. ун-т – № 2009101624/22; заявл. 19.01.2009; опубл. 10.10.2009. Бюл. № 28.
21. Ильницкий, И. И. Элементы автоматического управления процессами работы машин-станков / И. И. Ильницкий. Свердловск: УПИ, 1969.
22. Рейнгольдт, Ю. А. Электрооборудование и электроавтоматика подъемно-транспортных машин / Ю. А. Рейнгольдт. М.: Транспорт, 1968.
23. Сандлер, А. С. Электропривод и автоматизация металлорежущих станков / А. С. Сандлер. М.: Высшая школа, 1972.
24. Терехов, В. М. Системы управления электроприводов / В. М. Терехов, О. И. Осипов. М.: Академия, 2005.

25. Либерман, Я. Л. Исследование вертикальных колебаний груза на тросовой подвеске при резком торможении / Я. Л. Либерман, Ю. О. Лысый, К. Ю. Летнев // Современные технологии и бизнес: сб. научн. тр. Челябинск: ЧРО РЕАН, 2010.
26. Либерман, Я. Л. Адаптивное управление гашением колебаний груза при эксплуатации подъемно-транспортных машин / Я. Л. Либерман, К. Ю. Летнев, В. А. Штерензон // Современные проблемы машиностроения: тр. V Междун. научн.-техн. конф. Томск: НИ ТПУ, 2010.
27. Алексеев, С. П. Борьба с шумом и вибрацией в машиностроении / С. П. Алексеев, А. М. Казаков, Н. Н. Колотилов. М.: Машиностроение, 1970.
28. А.с. 1054284 СССР, МПК В66D 5/00. Способ торможения подъемной машины / А. Г. Степанов, Г. В. Клименков, В. Н. Шабалин, Ш. К. Яруллин; заявитель Пермский политех. ин-т – № 3409443/27-11; заявл. 01.03.82; опубл. 15.11.83. Бюл. № 42.
29. Ратмиров, В. А. Управление станками гибких производственных систем / В. А. Ратмиров. М.: Машиностроение, 1972.
30. Критерий приоритетного планирования мероприятий по профилактике производственного травматизма / А. А. Вершинин, Я. Л. Либерман, В. П. Ситников [и др.] // Изв. ВУЗов. Цветная металлургия. 1976. № 2.
31. Амортизационные расходы. Кран мостовой, стоимость / Горно-обога- тительное оборудование. URL: http://obogatitelnoe-oborudovanie.ru/amortizacionnye_rashody_kran_mostovoi_stoimost.html
32. Складенко, В. К. Расчет бюджета рабочего времени / В. К. Складенко // Справочник экономиста. 2006. №1.

ПРИЛОЖЕНИЕ

Программа «Система разгона – торможения управляемого привода»

// Drive Accel-Decel Control.cpp: главный файл проекта.

```
#include "stdafx.h"  
#include "DADC_Plot.h"  
#include "DADC_Help.h"  
#include "DADC.h"  
#include "DADC_Front.h"
```

```
using namespace DriveAccelDecelControl; //общее пространство имен
```

```
[STAThreadAttribute]  
int main(array<System::String ^> ^args)  
{  
    // Включение визуальных эффектов Windows XP до создания каких-либо  
    // элементов управления  
    Application::EnableVisualStyles();  
    Application::SetCompatibleTextRenderingDefault(false);  
  
    // Создание главного окна и его запуск  
    Application::Run(gcnew DADC_Front()); //запуск титульного окна  
    // программы  
    return 0;  
}
```

//файл DADC.h: класс формы - главного окна с функцией задания и расчета значений параметров

```
#pragma once
```

```
namespace DriveAccelDecelControl {
```

```
using namespace System;  
using namespace System::ComponentModel;  
using namespace System::Collections;  
using namespace System::Windows::Forms;  
using namespace System::Data;  
using namespace System::Drawing;
```

```
public ref class DADC : public System::Windows::Forms::Form
```

```

{
public:
DADC(void)
{
InitializeComponent();
InitializePlot(); //вызов функции инициализации объектов программы
}

protected:
~DADC()
{
if (components)
{
delete components;
}
delete pPlot; //удаление объектов программы при ее закрытии
delete Err_Graphics;
delete Ps_Prev;
delete DADC_File_Name;
delete DADC_Contents;
delete Ps;
delete Err_String;
delete Err_Image;
Application::Exit(); //выход из программы (закрывается скрытое титульное
окно программы)
}
private: System::Windows::Forms::Panel^ pnlMain;
private: System::Windows::Forms::GroupBox^ gbPlot;
private: System::Windows::Forms::Button^ bStop; //кнопка Стоп

private: System::Windows::Forms::Button^ bStart; //кнопка Старт

private: System::Windows::Forms::NumericUpDown^ nudSlowTime; //поле
Время замедления

private: System::Windows::Forms::NumericUpDown^ nudSlowRate; //поле
Замедленная скорость

private: System::Windows::Forms::NumericUpDown^ nudDecelTime; //поле
Время торможения

private: System::Windows::Forms::NumericUpDown^ nudWorkRate; //поле
Рабочая скорость

```

```

private: System::Windows::Forms::NumericUpDown^ nudAccelTime; //поле
Время разгона

private: System::Windows::Forms::Label^ lSlowTime;

private: System::Windows::Forms::Label^ lSlowRate;

private: System::Windows::Forms::Label^ lDecelTime;

private: System::Windows::Forms::Label^ lWorkRate;

private: System::Windows::Forms::Label^ lAccelTime;

private: System::ComponentModel::IContainer^ components;

private: System::Windows::Forms::Button^ bRestore; //кнопка Сбросить
private: System::Windows::Forms::Button^ bSave; //кнопка Сохранить

private: System::Windows::Forms::Timer^ tError;
private: System::Windows::Forms::Button^ bHelp; //кнопка Справка
private: System::Windows::Forms::Label^ lSlowRate_M;

private: System::Windows::Forms::Label^ lSlowTime_M;

private: System::Windows::Forms::Label^ lDecelTime_M;

private: System::Windows::Forms::Label^ lWorkRate_M;

private: System::Windows::Forms::Label^ lAccelTime_M;

private: System::Windows::Forms::PictureBox^ pbPlot; //объект отображения
графика

private: Plot^ pPlot; //объект графика
private: Params^ Ps; //объект параметров
private: System::String^ Err_String; //строка ошибки
private: Bitmap^ Err_Image; //объект рисунка при ошибке
private: Graphics^ Err_Graphics; //объект рисования при ошибке
private: Params^ Ps_Prev; //объект предыдущих параметров
private: System::String^ DADC_File_Name; //имя файла настроек
private: cli::array<System::String^,1>^ DADC_Contents; //параметры из
файла настроек

```

```

void InitializePlot() //инициализация объектов программы
{
DADC_File_Name = "DADC.dat"; //файл программы
DADC_Contents = gcnew
cli::array<System::String^,1>(6){ "0","0","0","0","0","0,0001"}; //массив
параметров со значениями по умолчанию
if (System::IO::File::Exists(DADC_File_Name))
DADC_Contents = System::IO::File::ReadAllLines(DADC_File_Name);
//считывание 6 строк файла в массив
nudAccelTime->Value = System::Convert::ToDecimal(DADC_Contents-
>GetValue(0)); //и установка значений полей...
DecPlaces_Define(nudAccelTime);
nudWorkRate->Value = System::Convert::ToDecimal(DADC_Contents-
>GetValue(1));
DecPlaces_Define(nudWorkRate);
nudDecelTime->Value = System::Convert::ToDecimal(DADC_Contents-
>GetValue(2));
DecPlaces_Define(nudDecelTime);
nudSlowTime->Value = System::Convert::ToDecimal(DADC_Contents-
>GetValue(3));
DecPlaces_Define(nudSlowTime);
nudSlowRate->Value = System::Convert::ToDecimal(DADC_Contents-
>GetValue(4));
DecPlaces_Define(nudSlowRate);
pPlot = gcnew Plot();
Ps = gcnew Params();
Ps_Prev = gcnew Params();
Ps->traz = float(nudAccelTime->Value); //и самих параметров
Ps->Vrab = float (nudWorkRate->Value);
Ps->ttr = float (nudDecelTime->Value);
Ps->Vzam = float (nudSlowRate->Value);
Ps->tzam = float (nudSlowTime->Value);
Ps->T = float(System::Convert::ToDecimal(DADC_Contents->GetValue(5)));
Ps_Prev->traz = float(nudAccelTime->Value);
Ps_Prev->Vrab = float (nudWorkRate->Value);
Ps_Prev->ttr = float (nudDecelTime->Value);
Ps_Prev->Vzam = float (nudSlowRate->Value);
Ps_Prev->tzam = float (nudSlowTime->Value);
Err_String = "";
Err_Image = gcnew Bitmap(pbPlot->Width, pbPlot->Height);
Err_Graphics = Graphics::FromImage(Err_Image);
}
private:

```

```

#pragma region Windows Form Designer generated code
/// <summary>
/// Обязательный метод для поддержки конструктора - не изменяйте
/// содержимое данного метода при помощи редактора кода.
/// </summary>
void InitializeComponent(void)
{
    this->components = (gcnew System::ComponentModel::Container());
    System::ComponentModel::ComponentResourceManager^ resources = (gcnew
    System::ComponentModel::ComponentResourceManager(DADC::typeid));
    this->pnlMain = (gcnew System::Windows::Forms::Panel());
    this->lSlowRate_M = (gcnew System::Windows::Forms::Label());
    this->lSlowTime_M = (gcnew System::Windows::Forms::Label());
    this->lDecelTime_M = (gcnew System::Windows::Forms::Label());
    this->lWorkRate_M = (gcnew System::Windows::Forms::Label());
    this->lAccelTime_M = (gcnew System::Windows::Forms::Label());
    this->bHelp = (gcnew System::Windows::Forms::Button());
    this->bRestore = (gcnew System::Windows::Forms::Button());
    this->bSave = (gcnew System::Windows::Forms::Button());
    this->lSlowTime = (gcnew System::Windows::Forms::Label());
    this->lSlowRate = (gcnew System::Windows::Forms::Label());
    this->lDecelTime = (gcnew System::Windows::Forms::Label());
    this->lWorkRate = (gcnew System::Windows::Forms::Label());
    this->lAccelTime = (gcnew System::Windows::Forms::Label());
    this->nudSlowTime = (gcnew System::Windows::Forms::NumericUpDown());
    this->nudSlowRate = (gcnew System::Windows::Forms::NumericUpDown());
    this->nudDecelTime = (gcnew System::Windows::Forms::NumericUpDown());
    this->nudWorkRate = (gcnew System::Windows::Forms::NumericUpDown());
    this->nudAccelTime = (gcnew System::Windows::Forms::NumericUpDown());
    this->bStop = (gcnew System::Windows::Forms::Button());
    this->bStart = (gcnew System::Windows::Forms::Button());
    this->gbPlot = (gcnew System::Windows::Forms::GroupBox());
    this->pbPlot = (gcnew System::Windows::Forms::PictureBox());
    this->tError = (gcnew System::Windows::Forms::Timer(this->components));
    this->pnlMain->SuspendLayout();
    (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
    >nudSlowTime))->BeginInit();
    (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
    >nudSlowRate))->BeginInit();
    (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
    >nudDecelTime))->BeginInit();
    (cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
    >nudWorkRate))->BeginInit();

```

```

(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nudAccelTime))->BeginInit();
this->gbPlot->SuspendLayout();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this->pbPlot))-
>BeginInit();
this->SuspendLayout();
//
// pnlMain
//
this->pnlMain->BorderStyle =
System::Windows::Forms::BorderStyle::Fixed3D;
this->pnlMain->Controls->Add(this->lSlowRate_M);
this->pnlMain->Controls->Add(this->lSlowTime_M);
this->pnlMain->Controls->Add(this->lDecelTime_M);
this->pnlMain->Controls->Add(this->lWorkRate_M);
this->pnlMain->Controls->Add(this->lAccelTime_M);
this->pnlMain->Controls->Add(this->bHelp);
this->pnlMain->Controls->Add(this->bRestore);
this->pnlMain->Controls->Add(this->bSave);
this->pnlMain->Controls->Add(this->lSlowTime);
this->pnlMain->Controls->Add(this->lSlowRate);
this->pnlMain->Controls->Add(this->lDecelTime);
this->pnlMain->Controls->Add(this->lWorkRate);
this->pnlMain->Controls->Add(this->lAccelTime);
this->pnlMain->Controls->Add(this->nudSlowTime);
this->pnlMain->Controls->Add(this->nudSlowRate);
this->pnlMain->Controls->Add(this->nudDecelTime);
this->pnlMain->Controls->Add(this->nudWorkRate);
this->pnlMain->Controls->Add(this->nudAccelTime);
this->pnlMain->Controls->Add(this->bStop);
this->pnlMain->Controls->Add(this->bStart);
this->pnlMain->Dock = System::Windows::Forms::DockStyle::Top;
this->pnlMain->Location = System::Drawing::Point(0, 0);
this->pnlMain->Name = L"pnlMain";
this->pnlMain->Size = System::Drawing::Size(1017, 72);
this->pnlMain->TabIndex = 0;
//
// lSlowRate_M
//
this->lSlowRate_M->AutoSize = true;
this->lSlowRate_M->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));

```

```

this->lSlowRate_M->Location = System::Drawing::Point(790, 45);
this->lSlowRate_M->Name = L"lSlowRate_M";
this->lSlowRate_M->Size = System::Drawing::Size(60, 16);
this->lSlowRate_M->TabIndex = 23;
this->lSlowRate_M->Text = L"усл.ед.";
//
// lSlowTime_M
//
this->lSlowTime_M->AutoSize = true;
this->lSlowTime_M->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->lSlowTime_M->Location = System::Drawing::Point(654, 45);
this->lSlowTime_M->Name = L"lSlowTime_M";
this->lSlowTime_M->Size = System::Drawing::Size(16, 16);
this->lSlowTime_M->TabIndex = 22;
this->lSlowTime_M->Text = L"с";
//
// lDecelTime_M
//
this->lDecelTime_M->AutoSize = true;
this->lDecelTime_M->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->lDecelTime_M->Location = System::Drawing::Point(517, 45);
this->lDecelTime_M->Name = L"lDecelTime_M";
this->lDecelTime_M->Size = System::Drawing::Size(16, 16);
this->lDecelTime_M->TabIndex = 21;
this->lDecelTime_M->Text = L"с";
//
// lWorkRate_M
//
this->lWorkRate_M->AutoSize = true;
this->lWorkRate_M->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->lWorkRate_M->Location = System::Drawing::Point(378, 45);
this->lWorkRate_M->Name = L"lWorkRate_M";
this->lWorkRate_M->Size = System::Drawing::Size(60, 16);
this->lWorkRate_M->TabIndex = 20;
this->lWorkRate_M->Text = L"усл.ед.";

```



```

//
// lAccelTime_M
//
this->lAccelTime_M->AutoSize = true;
this->lAccelTime_M->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->lAccelTime_M->Location = System::Drawing::Point(242, 45);
this->lAccelTime_M->Name = L"lAccelTime_M";
this->lAccelTime_M->Size = System::Drawing::Size(16, 16);
this->lAccelTime_M->TabIndex = 19;
this->lAccelTime_M->Text = L"с";
//
// bHelp
//
this->bHelp->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif",
8.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->bHelp->Image = (cli::safe_cast<System::Drawing::Image^ >(resources-
>GetObject(L"bHelp.Image")));
this->bHelp->Location = System::Drawing::Point(944, 6);
this->bHelp->Name = L"bHelp";
this->bHelp->Size = System::Drawing::Size(56, 56);
this->bHelp->TabIndex = 18;
this->bHelp->UseVisualStyleBackColor = true;
this->bHelp->Click += gcnew System::EventHandler(this,
&DADC::bHelp_Click);
//
// bRestore
//
this->bRestore->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 8.25F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->bRestore->Location = System::Drawing::Point(861, 38);
this->bRestore->Name = L"bRestore";
this->bRestore->Size = System::Drawing::Size(77, 23);
this->bRestore->TabIndex = 17;
this->bRestore->Text = L"Сбросить";
this->bRestore->UseVisualStyleBackColor = true;
this->bRestore->Click += gcnew System::EventHandler(this,
&DADC::bRestore_Click);

```



```

//
// bSave
//
this->bSave->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif",
8.25F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->bSave->Location = System::Drawing::Point(861, 10);
this->bSave->Name = L"bSave";
this->bSave->Size = System::Drawing::Size(77, 23);
this->bSave->TabIndex = 16;
this->bSave->Text = L"Сохранить";
this->bSave->UseVisualStyleBackColor = true;
this->bSave->Click += gcnew System::EventHandler(this,
&DADC::bSave_Click);
//
// lSlowTime
//
this->lSlowTime->AutoSize = true;
this->lSlowTime->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->lSlowTime->Location = System::Drawing::Point(578, 7);
this->lSlowTime->Name = L"lSlowTime";
this->lSlowTime->Size = System::Drawing::Size(125, 32);
this->lSlowTime->TabIndex = 13;
this->lSlowTime->Text = L"Время \r\nзамедления (tз)";
//
// lSlowRate
//
this->lSlowRate->AutoSize = true;
this->lSlowRate->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->lSlowRate->Location = System::Drawing::Point(715, 7);
this->lSlowRate->Name = L"lSlowRate";
this->lSlowRate->Size = System::Drawing::Size(112, 32);
this->lSlowRate->TabIndex = 14;
this->lSlowRate->Text = L"Замедленная \r\nскорость (Vз)";
//
// lDecelTime
//

```

```

this->lDecelTime->AutoSize = true;
this->lDecelTime->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->lDecelTime->ForeColor = System::Drawing::SystemColors::ControlText;
this->lDecelTime->Location = System::Drawing::Point(441, 7);
this->lDecelTime->Name = L"lDecelTime";
this->lDecelTime->Size = System::Drawing::Size(124, 32);
this->lDecelTime->TabIndex = 12;
this->lDecelTime->Text = L"Время \r\пторможения (тτ)";
//
// lWorkRate
//
this->lWorkRate->AutoSize = true;
this->lWorkRate->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->lWorkRate->Location = System::Drawing::Point(304, 7);
this->lWorkRate->Name = L"lWorkRate";
this->lWorkRate->Size = System::Drawing::Size(108, 32);
this->lWorkRate->TabIndex = 11;
this->lWorkRate->Text = L"Рабочая \r\пскорость (Vp)";
//
// lAccelTime
//
this->lAccelTime->AutoSize = true;
this->lAccelTime->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->lAccelTime->Location = System::Drawing::Point(167, 7);
this->lAccelTime->Name = L"lAccelTime";
this->lAccelTime->Size = System::Drawing::Size(96, 32);
this->lAccelTime->TabIndex = 10;
this->lAccelTime->Text = L"Время \r\празгона (tp)";
//
// nudSlowTime
//
this->nudSlowTime->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));

```

```

this->nudSlowTime->Location = System::Drawing::Point(578, 43);
this->nudSlowTime->Maximum = System::Decimal(gcnew cli::array<
System::Int32 >(4) { 1000, 0, 0, 0 });
this->nudSlowTime->Name = L"nudSlowTime";
this->nudSlowTime->Size = System::Drawing::Size(74, 22);
this->nudSlowTime->TabIndex = 3;
this->nudSlowTime->ValueChanged += gcnew System::EventHandler(this,
&DADC::nudSlowTime_ValueChanged);
this->nudSlowTime->Enter += gcnew System::EventHandler(this,
&DADC::nudSlowTime_Enter);
//
// nudSlowRate
//
this->nudSlowRate->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->nudSlowRate->Location = System::Drawing::Point(715, 43);
this->nudSlowRate->Maximum = System::Decimal(gcnew cli::array<
System::Int32 >(4) { 100000, 0, 0, 0 });
this->nudSlowRate->Name = L"nudSlowRate";
this->nudSlowRate->Size = System::Drawing::Size(74, 22);
this->nudSlowRate->TabIndex = 4;
this->nudSlowRate->ValueChanged += gcnew System::EventHandler(this,
&DADC::nudSlowRate_ValueChanged);
this->nudSlowRate->Enter += gcnew System::EventHandler(this,
&DADC::nudSlowRate_Enter);
//
// nudDecelTime
//
this->nudDecelTime->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->nudDecelTime->Location = System::Drawing::Point(441, 43);
this->nudDecelTime->Maximum = System::Decimal(gcnew cli::array<
System::Int32 >(4) { 1000, 0, 0, 0 });
this->nudDecelTime->Name = L"nudDecelTime";
this->nudDecelTime->Size = System::Drawing::Size(74, 22);
this->nudDecelTime->TabIndex = 2;
this->nudDecelTime->ValueChanged += gcnew System::EventHandler(this,
&DADC::nudDecelTime_ValueChanged);
this->nudDecelTime->Enter += gcnew System::EventHandler(this,
&DADC::nudDecelTime_Enter);

```

```

//
// nudWorkRate
//
this->nudWorkRate->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->nudWorkRate->Location = System::Drawing::Point(304, 43);
this->nudWorkRate->Maximum = System::Decimal(gcnew cli::array<
System::Int32 >(4) { 100000, 0, 0, 0 }));
this->nudWorkRate->Name = L"nudWorkRate";
this->nudWorkRate->Size = System::Drawing::Size(74, 22);
this->nudWorkRate->TabIndex = 1;
this->nudWorkRate->ValueChanged += gcnew System::EventHandler(this,
&DADC::nudWorkRate_ValueChanged);
this->nudWorkRate->Enter += gcnew System::EventHandler(this,
&DADC::nudWorkRate_Enter);
//
// nudAccelTime
//
this->nudAccelTime->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->nudAccelTime->Location = System::Drawing::Point(167, 43);
this->nudAccelTime->Maximum = System::Decimal(gcnew cli::array<
System::Int32 >(4) { 1000, 0, 0, 0 }));
this->nudAccelTime->Name = L"nudAccelTime";
this->nudAccelTime->Size = System::Drawing::Size(74, 22);
this->nudAccelTime->TabIndex = 0;
this->nudAccelTime->ValueChanged += gcnew System::EventHandler(this,
&DADC::nudAccelTime_ValueChanged);
this->nudAccelTime->Enter += gcnew System::EventHandler(this,
&DADC::nudAccelTime_Enter);
//
// bStop
//
this->bStop->FlatStyle = System::Windows::Forms::FlatStyle::Popup;
this->bStop->Image = (cli::safe_cast<System::Drawing::Image^ >(resources-
>GetObject(L"bStop.Image")));
this->bStop->ImageAlign = System::Drawing::ContentAlignment::TopCenter;
this->bStop->Location = System::Drawing::Point(92, 6);
this->bStop->Name = L"bStop";
this->bStop->Size = System::Drawing::Size(56, 56);

```

```

this->bStop->TabIndex = 20;
this->bStop->UseVisualStyleBackColor = true;
this->bStop->Click += gcnw System::EventHandler(this,
&DADC::bStop_Click);
//
// bStart
//
this->bStart->FlatStyle = System::Windows::Forms::FlatStyle::Popup;
this->bStart->Image = (cli::safe_cast<System::Drawing::Image^ >(resources-
>GetObject(L"bStart.Image")));
this->bStart->ImageAlign = System::Drawing::ContentAlignment::TopCenter;
this->bStart->Location = System::Drawing::Point(13, 6);
this->bStart->Name = L"bStart";
this->bStart->Size = System::Drawing::Size(56, 56);
this->bStart->TabIndex = 19;
this->bStart->UseVisualStyleBackColor = true;
this->bStart->Click += gcnw System::EventHandler(this,
&DADC::bStart_Click);
//
// gbPlot
//
this->gbPlot->Controls->Add(this->pbPlot);
this->gbPlot->Dock = System::Windows::Forms::DockStyle::Fill;
this->gbPlot->Font = (gcnw System::Drawing::Font(L"Microsoft Sans Serif",
9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->gbPlot->Location = System::Drawing::Point(0, 72);
this->gbPlot->Name = L"gbPlot";
this->gbPlot->Size = System::Drawing::Size(1017, 194);
this->gbPlot->TabIndex = 1;
this->gbPlot->TabStop = false;
this->gbPlot->Text = L"График разгона и торможения";
//
// pbPlot
//
this->pbPlot->Dock = System::Windows::Forms::DockStyle::Fill;
this->pbPlot->Location = System::Drawing::Point(3, 18);
this->pbPlot->Name = L"pbPlot";
this->pbPlot->Size = System::Drawing::Size(1011, 173);
this->pbPlot->TabIndex = 0;
this->pbPlot->TabStop = false;
this->pbPlot->Paint += gcnw
System::Windows::Forms::PaintEventHandler(this, &DADC::pbPlot_Paint);

```

```

//
// tError
//
this->tError->Interval = 500;
this->tError->Tick += gcnew System::EventHandler(this,
&DADC::tError_Tick);
//
// DADC
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(1017, 266);
this->Controls->Add(this->gbPlot);
this->Controls->Add(this->pnlMain);
this->DoubleBuffered = true;
this->MaximizeBox = false;
this->Name = L"DADC";
this->Text = L"Система разгона–торможения управляемого привода";
this->WindowState =
System::Windows::Forms::FormWindowState::Maximized;
this->pnlMain->ResumeLayout(false);
this->pnlMain->PerformLayout();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nudSlowTime))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nudSlowRate))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nudDecelTime))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nudWorkRate))->EndInit();
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this-
>nudAccelTime))->EndInit();
this->gbPlot->ResumeLayout(false);
(cli::safe_cast<System::ComponentModel::ISupportInitialize^ >(this->pbPlot))-
>EndInit();
this->ResumeLayout(false);

}
#pragma endregion

private: System::Void bStart_Click(System::Object^ sender,
System::EventArgs^ e) { //функция кнопки Старт
if (bStart->FlatStyle != System::Windows::Forms::FlatStyle::Standard)
return; //если стиль кнопки нестандартный, то не запускать процесс

```



```

//(используется как ограничитель запуска при еще не сохраненных
параметрах)
pPlot->Reset(); //подготовка графика к запуску
pPlot->InitGraphics(pbPlot->Width,pbPlot->Height,30,float(pbPlot->Height-
30),float(pbPlot->Width-30),30); //инициализация графика
bStart->FlatStyle = System::Windows::Forms::FlatStyle::Flat; //установка
кнопок в состояние ожидания завершения процесса разгона
this->TopMost = true;
this->ControlBox = false;
bSave->Enabled = false;
bRestore->Enabled = false;
pPlot->Start(true, true, true); //запуск разгона
this->TopMost = false; //установка кнопок в состояние ожидания
начала/завершения процесса торможения
this->ControlBox = true;
bStart->FlatStyle = System::Windows::Forms::FlatStyle::Popup;
bStop->FlatStyle = System::Windows::Forms::FlatStyle::Standard;
}

```

```

private: System::Void bStop_Click(System::Object^ sender,
System::EventArgs^ e) { //функция кнопки Стоп
    if (bStop->FlatStyle != System::Windows::Forms::FlatStyle::Standard)
return; //по кнопкам то же, что и для кнопки Старт
    bStop->FlatStyle = System::Windows::Forms::FlatStyle::Flat;
    this->TopMost = true;
    this->ControlBox = false;
    pPlot->Start(true, true, true); //запуск процесса торможения
    this->TopMost = false;
    this->ControlBox = true;
    bStop->FlatStyle = System::Windows::Forms::FlatStyle::Popup;
    bStart->FlatStyle = System::Windows::Forms::FlatStyle::Standard;
    bSave->Enabled = true;
    bRestore->Enabled = true;
}

```

```

private: System::Void nudAccelTime_ValueChanged(System::Object^ sender,
System::EventArgs^ e) {
    DecPlaces_Define(nudAccelTime);
}
private: System::Void nudWorkRate_ValueChanged(System::Object^ sender,
System::EventArgs^ e) {
    DecPlaces_Define(nudWorkRate);
}

```

```
private: System::Void nudDecelTime_ValueChanged(System::Object^ sender,
System::EventArgs^ e) {
    DecPlaces_Define(nudDecelTime);
}
```

```
private: System::Void nudSlowRate_ValueChanged(System::Object^ sender,
System::EventArgs^ e) {
    DecPlaces_Define(nudSlowRate);
}
```

```
private: System::Void nudSlowTime_ValueChanged(System::Object^ sender,
System::EventArgs^ e) {
    DecPlaces_Define(nudSlowTime);
}
```

```
private: void DecPlaces_Define (System::Windows::Forms::NumericUpDown^
nud) //функция изменения количества знаков после запятой в полях
{
    System::String^ s = nud->Value.ToString();
    int i=0;
    if (s->Contains(","))
        i = s->Length-s->IndexOf(",")-1;
    else if (s->Contains("."))
        i = s->Length-s->IndexOf(".")-1;
    if (i!=nud->DecimalPlaces)
    {
        nud->DecimalPlaces = i;
        nud->Increment=System::Decimal(System::Math::Pow(10,i*(-1)));
    }
    delete s;
}
```

```
private: System::Void bRestore_Click(System::Object^ sender,
System::EventArgs^ e) { //функция кнопки Сбросить
    nudAccelTime->Value = System::Decimal(Ps_Prev->traz); //восстановление
параметров до предыдущих
    nudWorkRate->Value = System::Decimal(Ps_Prev->Vrab);
    nudDecelTime->Value = System::Decimal(Ps_Prev->ttr);
    nudSlowRate->Value = System::Decimal(Ps_Prev->Vzam);
    nudSlowTime->Value = System::Decimal(Ps_Prev->tzam);
    Ps->traz=Ps_Prev->traz;
    Ps->Vrab=Ps_Prev->Vrab;
    Ps->ttr=Ps_Prev->ttr;
    Ps->Vzam=Ps_Prev->Vzam;
    Ps->tzam=Ps_Prev->tzam;
}
```



```

private: System::Void bSave_Click(System::Object^ sender,
System::EventArgs^ e) { //функция кнопки Сохранить
bStart->FlatStyle = System::Windows::Forms::FlatStyle::Popup;
delete pPlot; //удаляем предыдущий объект графика...
pPlot = gcnew Plot(); //и создаем новый
pbPlot->Refresh(); //стираем изображение

pPlot->Draw_Object = pbPlot; //установка объекта рисования для графика

Ps_Prev->traz = Ps->traz; //сохранение старых параметров в объект
предыдущих
Ps_Prev->Vrab = Ps->Vrab;
Ps_Prev->ttr = Ps->ttr;
Ps_Prev->Vzam = Ps->Vzam;
Ps_Prev->tzam = Ps->tzam;

Ps->traz = float(nudAccelTime->Value); //загрузка из полей значений в
объект параметров
Ps->Vrab = float (nudWorkRate->Value);
Ps->ttr = float (nudDecelTime->Value);
Ps->Vzam = float (nudSlowRate->Value);
Ps->tzam = float (nudSlowTime->Value);
//Ps->T = (Ps->traz + Ps->ttr + (Ps->traz + Ps->ttr)/10)/100;
//Ps->T = float(0.0001);

//Создание и вывод объекта ошибки при ошибочных значениях
параметров
Err_String = "";
delete Err_Image;
delete Err_Graphics;
float tl = Ps->T*5;
if (Ps->traz<=tl)
Err_String = "Время разгона (tr) указывается в секундах (с) \ни не может
быть равным нулю!!!";
else if (Ps->Vrab<=tl)
Err_String = "Рабочая скорость (Vp) указывается в условных единицах
(усл.ед.) \ни не может быть равной нулю!!!";
else if (Ps->ttr<=tl)
Err_String = "Время торможения (tt) указывается в секундах (с),
\пвключает в себя время замедления (tz) \ни не может быть равным
нулю!!!";
else if ((Ps->ttr-Ps->tzam)<=(tl*2))
Err_String = "Время работы двигателя при торможении на скорости
замедленного хода (tz) \пказывается в секундах (с), составляет часть

```

```

времени торможения (tt), \не может быть больше tt, но может быть
равной нулю!!!";
else if ((Ps->tzam>=Ps->T)&&((Ps->Vzam> (5*Ps->Vrab/6 + Ps->T*((-4)*Ps-
>Vrab/6) / ((Ps->ttor-Ps->tzam)/5))))||((Ps->Vzam< (Ps->Vrab/6-Ps->T*((-4)*Ps-
>Vrab/6) / ((Ps->ttor-Ps->tzam)/5))))))
Err_String = "Скорость замедленного хода (Vз) указывается в условных
единицах (усл.ед.)\ни должна находиться в пределах между V1 = " +
float(Ps->Vrab/6-Ps->T*((-4)*Ps->Vrab/6) / ((Ps->ttor-Ps-
>tzam)/5)).ToString() + " и V2 = " + float(5*Ps->Vrab/6 + Ps->T*((-4)*Ps-
>Vrab/6) / ((Ps->ttor-Ps->tzam)/5)).ToString() + ", \т.е. эта скорость может
быть установлена лишь на участке торможения \с максимальным
замедлением!!!";
if (Err_String->Length>0) //если объект ошибки был создан
{
bSave->Enabled=false;
Err_Image = gcnw Bitmap(pbPlot->Width,pbPlot->Height);
Err_Graphics = Graphics::FromImage(Err_Image);
Err_Graphics->DrawString (Err_String,gcnw System::Drawing::Font("Arial",
18), gcnw SolidBrush(Color::Red),20,20);
pbPlot->Refresh();
tError->Enabled=true; //то выводим его на экран
return; //и выходим из функции, не сохранив параметры
}

//Далее рассчитываем и устанавливаем параметры
DADC_Contents->SetValue(Ps->traz.ToString(),0);
DADC_Contents->SetValue(Ps->Vrab.ToString(),1);
DADC_Contents->SetValue(Ps->ttor.ToString(),2);
DADC_Contents->SetValue(Ps->tzam.ToString(),3);
DADC_Contents->SetValue(Ps->Vzam.ToString(),4);
DADC_Contents->SetValue(Ps->T.ToString(),5);
System::IO::File::WriteAllLines(DADC_File_Name,DADC_Contents);
//сохранение в файл

if (Ps->Vzam < Ps->T) //если скорость замедления равна нулю (возможно
только если время замедления равно нулю)
Ps->Vzam = Ps->Vrab/2;

//Устанавливаем параметры участков графика
pPlot->Name = "График разгона и торможения";
pPlot->Vmax = Ps->Vrab;
//pPlot->Vmax = 3500;
pPlot->tsum = Ps->traz + Ps->ttor + (Ps->traz + Ps->ttor)/10;
//pPlot->tsum = Ps->traz + Ps->ttor + (Ps->traz + Ps->ttor)/2;

```

```
pPlot->P = Ps->T;
```

```
Segment^ st; //объект состояния (стадии)
```

```
Segment^ sg; //объект сегмента (участка)
```

```
//СОСТОЯНИЕ 1
```

```
pPlot->States->Add(gcnew Segment()); //добавляем состояние в объект  
графика
```

```
st = pPlot->States[0];
```

```
st->Name = "Разгон"; //инициализируем его параметры
```

```
st->Symbol = "tp";
```

```
st->Vmax = Ps->Vrab;
```

```
st->tend = Ps->traz;
```

```
//Сегмент 1
```

```
st->Segments->Add(gcnew Segment()); //добавляем сегмент (участок) в  
объект графика
```

```
sg = st->Segments[0];
```

```
sg->Name = "1 ст. разгона"; //инициализируем его параметры
```

```
sg->Symbol = "tp1";
```

```
sg->Vmax = Ps->Vrab/6;
```

```
sg->tend = 2*Ps->traz/5;
```

```
sg->a = (Ps->Vrab/6) / (2*Ps->traz/5);
```

```
//Сегмент 2
```

```
st->Segments->Add(gcnew Segment());
```

```
sg = st->Segments[1];
```

```
sg->Name = "2 ст. разгона";
```

```
sg->Symbol = "tp2";
```

```
sg->Vmax = 5*Ps->Vrab/6;
```

```
sg->tend = Ps->traz/5;
```

```
sg->a = (4*Ps->Vrab/6) / (Ps->traz/5);
```

```
//Сегмент 3
```

```
st->Segments->Add(gcnew Segment());
```

```
sg = st->Segments[2];
```

```
sg->Name = "3 ст. разгона";
```

```
sg->Symbol = "tp3";
```

```
sg->Vmax = Ps->Vrab;
```

```
sg->tend = 2*Ps->traz/5;
```

```
sg->a = (Ps->Vrab/6) / (2*Ps->traz/5);
```

//СОСТОЯНИЕ 2

```
pPlot->States->Add(gcnew Segment());  
st = pPlot->States[1];
```

```
st->Name = "Работа";  
st->Symbol = "Vpaб";  
st->tend = (Ps->traz + Ps->ttor)/10;  
//st->tend = (Ps->traz + Ps->ttor)/2;
```

//Сегмент 1

```
st->Segments->Add(gcnew Segment());  
sg = st->Segments[0];
```

```
sg->Name = "Работа";  
sg->Symbol = "Vpaб";  
sg->tend = (Ps->traz + Ps->ttor)/10;  
//sg->tend = (Ps->traz + Ps->ttor)/2;  
sg->Vmax = Ps->Vrab;  
sg->a = 0;
```

//СОСТОЯНИЕ 3

```
pPlot->States->Add(gcnew Segment());  
st = pPlot->States[2];
```

```
st->Name = "Торможение";  
st->Symbol = "V0";  
st->Vmax = 0;  
st->tend = Ps->ttor;
```

//Сегмент 1

```
st->Segments->Add(gcnew Segment());  
sg = st->Segments[0];
```

```
sg->Name = "1 ст. торможения";  
sg->Symbol = "т1";  
sg->Vmin = 5*Ps->Vrab/6;  
sg->tbeg = -1;  
sg->tend = 2*(Ps->ttor-Ps->tzam)/5;  
sg->a = (-Ps->Vrab/6) / (2*(Ps->ttor-Ps->tzam)/5);
```

//Сегмент 2

```
st->Segments->Add(gcnew Segment());
```

```

sg = st->Segments[1];

sg->Name = "2 ст. торможения";
sg->Symbol = "т2";
sg->Vmin = Ps->Vzam;
sg->tend = (Ps->Vzam-(5*Ps->Vrab/6))/((-4)*Ps->Vrab/6) / ((Ps->ttor-Ps->tzam)/5));
sg->a = ((-4)*Ps->Vrab/6) / ((Ps->ttor-Ps->tzam)/5);

```

//Сегмент 3

```

st->Segments->Add(gcnew Segment());
sg = st->Segments[2];

```

```

sg->Name = "Замедление";
sg->Symbol = "V3";
sg->tend = Ps->tzam;
sg->a = 0;

```

//Сегмент 4

```

st->Segments->Add(gcnew Segment());
sg = st->Segments[3];

```

```

sg->Name = "3 ст. торможения";
sg->Symbol = "т3";
sg->Vmin = Ps->Vrab/6;
sg->tend = ((Ps->Vrab/6)-Ps->Vzam) / (((-4)*Ps->Vrab/6)/((Ps->ttor-Ps->tzam)/5));
sg->a = ((-4)*Ps->Vrab/6) / ((Ps->ttor-Ps->tzam)/5);

```

//Сегмент 5

```

st->Segments->Add(gcnew Segment());
sg = st->Segments[4];

```

```

sg->Name = "4 ст. торможения";
sg->Symbol = "т4";
sg->tend = 2*(Ps->ttor-Ps->tzam)/5;
sg->Vmin = 0;
sg->a = (-Ps->Vrab/6) / (2*(Ps->ttor-Ps->tzam)/5);

```

```

pPlot->Reset();

```

//Создаем объект легенды – текстовое обозначение для участков графика и его расположение

```

Legend^ lg;

```

```
pPlot->X->Symbol = "t,c";
pPlot->Y->Symbol = "V,усл.ед.";
pPlot->X->Legend_Levels = 2;
pPlot->Y->Legend_Levels = 2;
```

```
pPlot->X->Legends->Add(gcnew Legend());
lg=pPlot->X->Legends[0];
lg->Name = "Время разгона";
lg->Symbol = "tp";
lg->Show_Level = 2;
lg->Beg = 0;
lg->End = Ps->traz;
```

```
pPlot->X->Legends->Add(gcnew Legend());
lg=pPlot->X->Legends[1];
lg->Name = "Время торможения";
lg->Symbol = "tt";
lg->Show_Level = 2;
//lg->Beg = Ps->traz + (Ps->traz + Ps->ttor)/2;
lg->Beg = Ps->traz + (Ps->traz + Ps->ttor)/10;
lg->End = lg->Beg + Ps->ttor;
```

```
pPlot->X->Legends->Add(gcnew Legend());
lg=pPlot->X->Legends[2];
lg->Name = "Время замедления";
lg->Symbol = "tz";
if (Ps->tzam>Ps->T)
    lg->Show_Level = 1;
lg->Beg = Ps->traz + (Ps->traz + Ps->ttor)/10 + 2*(Ps->ttor-Ps->tzam)/5 + (Ps-
>Vzam-(5*Ps->Vrab/6))/((-4)*Ps->Vrab/6) / ((Ps->ttor-Ps->tzam)/5));
//lg->Beg = Ps->traz + (Ps->traz + Ps->ttor)/2 + 2*(Ps->ttor-Ps->tzam)/5 + (Ps-
>Vzam-(5*Ps->Vrab/6))/((-4)*Ps->Vrab/6) / ((Ps->ttor-Ps->tzam)/5));
lg->End = lg->Beg + Ps->tzam;
```

```
pPlot->Y->Legends->Add(gcnew Legend());
lg=pPlot->Y->Legends[0];
lg->Name = "Рабочая скорость";
lg->Symbol = "Vp";
lg->Show_Level = 2;
lg->Beg = 0;
lg->End = Ps->Vrab;
```

```
pPlot->Y->Legends->Add(gcnew Legend());
```

```

lg=pPlot->Y->Legends[1];
lg->Name = "Замедленная скорость";
lg->Symbol = "V3";
lg->Show_Level = 1;
lg->Beg = 0;
lg->End = Ps->Vzam;

pPlot->InitGraphics(pbPlot->Width,pbPlot->Height,30,float(pbPlot->Height-
30),float(pbPlot->Width-30),30); //инициализируем графику
pPlot->Start(true, true, false); //запускаем только прорисовку левой части
графика
pPlot->Start(true, true, false); //запускаем только прорисовку правой части
графика
bStart->FlatStyle = System::Windows::Forms::FlatStyle::Standard;
}

private: System::Void tError_Tick(System::Object^ sender,
System::EventArgs^ e) { //таймер объекта ошибки
static int i=6;
static bool show = false;
if (show)
{
Err_Graphics->DrawString (Err_String,gcnew System::Drawing::Font("Arial",
18), gcnew SolidBrush(Color::Red),20,20);
pbPlot->Refresh();
}
else
{
Err_Graphics->Clear(Color::Transparent);
pbPlot->Refresh();
}
show = !show;
i--;
if (i==0)
{
tError->Enabled = false;
bSave->Enabled=true;
i=6;
show = false;
}

}

```

```

private: System::Void nudAccelTime_Enter(System::Object^ sender,
System::EventArgs^ e) {
    nudAccelTime->Select(0, nudAccelTime->ToString()->Length); //выбор
текста при входе в поле
}
private: System::Void nudWorkRate_Enter(System::Object^ sender,
System::EventArgs^ e) {
    nudWorkRate->Select(0, nudWorkRate->ToString()->Length);

}
private: System::Void nudDecelTime_Enter(System::Object^ sender,
System::EventArgs^ e) {
    nudDecelTime->Select(0, nudDecelTime->ToString()->Length);

}
private: System::Void nudSlowTime_Enter(System::Object^ sender,
System::EventArgs^ e) {
    nudSlowTime->Select(0, nudSlowTime->ToString()->Length);

}
private: System::Void nudSlowRate_Enter(System::Object^ sender,
System::EventArgs^ e) {
    nudSlowRate->Select(0, nudSlowRate->ToString()->Length);

}
private: System::Void bHelp_Click(System::Object^ sender,
System::EventArgs^ e) { //функция кнопки Справка
    System::Windows::Forms::Form^ dadc_help = gcnew DADC_Help();
    dadc_help->Show();
}
private: System::Void pbPlot_Paint(System::Object^ sender,
System::Windows::Forms::PaintEventArgs^ e) {
    //графический вывод в область объекта отображения графика или ошибки
    if (Err_String->Length==0)
        e->Graphics->DrawImage(pPlot->B,0,0);
    else
        e->Graphics->DrawImage(Err_Image,0,0);
}
};
}

//файл DADC_Plot.h: класс обработки заданных участков с выводом на
экран в виде графика и функцией вывода на привод
#pragma once

```



```

using namespace System::Collections::Generic;
using namespace System::Drawing;

public ref class Legend //класс легенды - текстовой информации для
участков и ее расположения
{
public:
System::String^ Name;
System::String^ Symbol; //отображаемый символ
int Show_Level; //уровень расположения
float Beg; //начало и конец выносных стрелок
float End;
Legend(){Name="";Symbol="";Show_Level=0;Beg=0;End=1;}
~Legend(){delete Name;delete Symbol;}
};

public ref class Segment //класс участка или стадии
{
public:
System::String^ Name;
System::String^ Symbol;
float Vmin; //минимальная скорость на участке
float Vmax; //максимальная скорость на участке
float tbeg; //время начальное участка (обычно 0)
float tend; //время окончания (временной промежутков) участка
float a; //ускорение на участке
List<Segment^>^ Segments;
Segment(){Name="";Symbol="";Vmin=-1;Vmax=-1;tbeg=0;tend=-
1;a=0;Segments=gcnnew List<Segment^>();}
~Segment(){delete Name;delete Symbol; delete Segments;}
};

public ref class Coordinate //класс координаты для отображения графика на
экране
{
public:
float Scale; //масштаб
float Beg; //начало оси
float End; //окончание оси
int Reverse; //переворот оси
System::String^ Symbol;
int Legend_Levels; //количество уровней отображения легенд
List<Legend^>^ Legends;

```

```

Coordinate(){Scale=1;Beg=0;End=0;Reverse=1;Symbol="";Legend_Levels=0;
Legends=gcnew List<Legend^>();}
~Coordinate(){delete Symbol; delete Legends;}
};

```

```

public ref class Plot //главный класс графика
{
public:
System::String^ Name;
float t; //время
float V; //скорость
System::Diagnostics::Stopwatch^ sw; //таймер реального времени
float tsum; //суммарное время
float Vmax; //максимальная скорость
float P; //период вывода скорости на привод
float _P; //счетчик периода
float T; //дельта  $t$  – время между двумя циклами
float tloc; //локальное время на участке
bool Enabled; //производить ли расчет параметров графика
bool output; //выводить ли скорость на привод
bool draw; //выводить ли график на экран
List<Segment^>^ States; //стадии
List<Segment^>::Enumerator^ State_Enum; //итератор по стадиям
List<Segment^>::Enumerator^ Segment_Enum; //итератор по участкам
Segment^ State_Current; //текущая стадия
Segment^ Segment_Current; //текущий участок
Coordinate^ X; //координаты
Coordinate^ Y;
Bitmap^ B; //объект изображения графика
Graphics^ G; //объект рисования на изображении
Pen^ Axes_Pen; //объекты перьев для осей, графика, выносных линий и
штрихпунктирных линий
Pen^ Graph_Pen;
Pen^ Measure_Pen;
Pen^ Dotted_Pen;
Font^ Plot_Font; //шрифт графика
SolidBrush^ Plot_Brush; //кисть графика
System::Windows::Forms::PictureBox^ Draw_Object; //внешний объект
изображения на форме

void InitGraphics(int, int, float, float, float, float); //функция инициализации
графики

```

```

void ResetGraphics(Pen^ _Axes_Pen, Pen^ _Graph_Pen, Pen^ _Measure_Pen,
Pen^ _Dotted_Pen, SolidBrush^ _Plot_Brush, Font^ _Plot_Font); //функция
сбрасывания графики
void Reset(); //функция сбрасывания параметров
void Start(bool, bool, bool); //запуск прохода по участкам с выводом
void State_Next(); //следующая стадия
void Segment_Next(); //следующий участок
bool Update(); //функция прохода по участкам
bool Output(); //функция вывода скорости на привод
void Draw(float, float, float, float, System::Drawing::Pen^, bool); //прорисовка
линий
void Draw_String(float, float, System::String^,Font^,Brush^, bool);
//прорисовка надписей

```

```

Plot(){Name="";t=0;V=0;tsum=0;Vmax=0;P=0;_P=0;tloc=0;Enabled=false;out
put=false; draw=true;States=gcnew List<Segment^>();State_Enum=States-
>GetEnumerator();Segment_Enum=nullptr;State_Current=nullptr;Segment_Cur
rent=nullptr;X=gcnew Coordinate();Y=gcnew Coordinate();B = nullptr;G =
nullptr;Axes_Pen=nullptr;Graph_Pen=nullptr;Measure_Pen=nullptr;Dotted_Pen
=nullptr;ResetGraphics(gcnew Pen (Color::Red,2.5),gcnew Pen
(Color::Blue,2),gcnew Pen (Color::Indigo,1.5), gcnew Pen (Color::HotPink,0.5),
gcnew SolidBrush(Color::Black), gcnew Font("Arial", 14));InitGraphics(1, 1, 0,
0, 1, 1);Draw_Object=nullptr;sw = gcnew System::Diagnostics::Stopwatch();}
~Plot(){delete Name; delete States; delete State_Enum; delete Segment_Enum;
delete State_Current; delete Segment_Current; delete X; delete Y; delete B;
delete G; delete Axes_Pen; delete Graph_Pen; delete Measure_Pen; delete
Dotted_Pen; delete Plot_Font; delete Plot_Brush;delete sw;}
};

```

```

bool Plot::Update() //функция прохода по участкам
{
if (!Enabled) return false; //выходит из функции, если обход участков
завершен (устанавливает функция Segment_Next) или программа по другой
причине остановила расчет параметров
float x1, y1, x2, y2;
sw->Stop(); //остановка таймера
if (output) //установка времени, прошедшего между циклами (если вывод
на привод осуществляется)
T = float(sw->Elapsed.TotalSeconds);
sw->Reset(); //сброс таймера
sw->Start(); //запуск заново
x1=t; //для прорисовки первой точки линии в x1 устанавливаем старое
значение времени

```

```

y1=V; //в y1 – старое значение скорости
t+=T; //теперь изменяем времена, период, скорость и координаты второй
точки в соответствии с T
tloc+=T;
_P-=T;
V+=(T*Segment_Current->a);
x2=t;
y2=V;

```

```

if (((Segment_Current->tend >0)&&(tloc>=Segment_Current-
>tend))||((Segment_Current->Vmax>0)&&(V>=Segment_Current->Vmax))||
((Segment_Current->Vmin>=0)&&(V<=Segment_Current->Vmin))) //если
участок завершился (по времени или скорости)
{
if (Segment_Current->tend >0) //подгоняем время для корректного вывода
макс/мин скорости
t = t + (Segment_Current->tend-tloc);
if ((Segment_Current->Vmax>0)&&(V>=Segment_Current->Vmax))
//корректно выводим макс/мин скорость
V = Segment_Current->Vmax;
else if ((Segment_Current->Vmin>=0)&&(V<=Segment_Current->Vmin))
V = Segment_Current->Vmin;
x2=t;
y2=V;
_P=0;
Output(); //выводим на привод
Draw(x1,y1,x2,y2, Graph_Pen, true); //и на экран
Segment_Next(); //следующий участок
}

```

```

else //если участок еще не завершился, то...
{
Output(); //просто выводим на привод
Draw(x1,y1,x2,y2, Graph_Pen, true); //и на экран
}
return true; //возвращаем удачное завершение
}

```

```

void Plot::Start(bool _graph, bool _draw, bool _output) //запуск прохода по
участкам с выводом
{
Enabled = _graph;
draw = _draw;
output = _output;

```

```

_P=0;
T=1;
Output(); //инициализация привода
_P=P;
sw->Reset();
sw->Start(); //запуск таймера
while (Update()); //последовательный проход по всем участкам
}

```

```

void Plot::State_Next() //следующая стадия
{
State_Current = nullptr;
if (State_Enum->MoveNext()) //переход на следующую стадию
State_Current = State_Enum->Current;
}

```

```

void Plot::Segment_Next() //следующий участок
{
tloc = 0; //обнуление времени участка
Segment_Current = nullptr;
while ((Segment_Current == nullptr) && Segment_Enum->MoveNext())
//переход на следующий участок
{
if (Segment_Enum->Current->tend>0) //если время на нем не равно нулю
Segment_Current = Segment_Enum->Current;
}
if (Segment_Current == nullptr) //если больше участков на стадии нет
{
State_Current = nullptr;
while ((Segment_Current == nullptr) && State_Enum->MoveNext())
{
State_Current = State_Enum->Current;
Segment_Enum = State_Current->Segments->GetEnumerator();
while ((Segment_Current == nullptr) && Segment_Enum->MoveNext())
{ //переходим на следующую стадию, если она имеется (с хотя бы одним
участком со временем больше нуля)
if (Segment_Enum->Current->tend>0)
Segment_Current = Segment_Enum->Current;
}
}
}
if (Segment_Current == nullptr) //если больше участков нет
{

```

```

Enabled = false; //останавливаем процесс обхода участков/вывода скорости
State_Current = nullptr;
}
else
{
if (Segment_Current->tbeg < 0) // также временно останавливаем, если
начальное время меньше нуля (условное обозначение для ожидания
продолжения вывода после команды пользователя или программы
(например, при нажатии на кнопку "Стоп")
Enabled = false;
else //или продолжаем работу
Enabled = true;
}
}

void Plot::Reset() //функция сбрасывания параметров
{
Enabled = false;
V = 0;
t = 0;
sw->Reset();
State_Current = nullptr;
Segment_Current = nullptr;
State_Enum = States->GetEnumerator();
State_Next();
if (State_Current != nullptr)
{
Segment_Enum = State_Current->Segments->GetEnumerator();
Segment_Next();
}
}

bool Plot::Output() //функция вывода скорости на привод
{
if((output) && (_P<=0) && (Segment_Current->a!=0))
{//Вывод скорости V на подключенный контроллер (если нужно
(output=true), с указанным периодом и при изменении скорости)
//Здесь нужно поместить код работы с конкретным контроллером и
приводом
_P=P; //устанавливаем счетчик периода заново
}
return true;
}

```

```

void Plot::Draw (float X1, float Y1, float X2, float Y2, System::Drawing::Pen^
pen, bool self)
{ //прорисовка линий
if (draw) //если необходимо
{ G->DrawLine(pen,X->Beg + X1/X->Scale*X->Reverse, Y->Beg + Y1/Y-
>Scale*Y->Reverse, X->Beg + X2/X->Scale*X->Reverse, Y->Beg + Y2/Y-
>Scale*Y->Reverse);
if (self) //если объект Plot сам производит вывод в окно (может также
производиться в классе формы)
Draw_Object->Refresh();
}
}

```

```

void Plot::Draw_String(float X0, float Y0, System::String^ str,Font^
font,Brush^ brush, bool self)
{ //прорисовка текста
if (draw) //аналогично предыдущему
{
G->DrawString(str,font,brush,X->Beg + X0/X->Scale*X->Reverse,Y->Beg +
Y0/Y->Scale*Y->Reverse);
if (self)
Draw_Object->Refresh();
}
}

```

```

void Plot::ResetGraphics(Pen^ _Axes_Pen, Pen^ _Graph_Pen, Pen^
_Measure_Pen, Pen^ _Dotted_Pen, SolidBrush^ _Plot_Brush, Font^
_Plot_Font)
{ //функция сбрасывания графики

```

```

Axes_Pen = _Axes_Pen;
Axes_Pen->EndCap = Drawing2D::LineCap::ArrowAnchor;
Graph_Pen = _Graph_Pen;
Graph_Pen->EndCap = Drawing2D::LineCap::NoAnchor;
Measure_Pen = _Measure_Pen;
Measure_Pen->StartCap = Drawing2D::LineCap::ArrowAnchor;
Measure_Pen->EndCap = Drawing2D::LineCap::ArrowAnchor;
Dotted_Pen = _Dotted_Pen;
Dotted_Pen->DashPattern = gcnew cli::array<float,1> {2.0,2.0,2.0,2.0};
Plot_Brush = _Plot_Brush;
Plot_Font = _Plot_Font;

}

```

```

void Plot::InitGraphics(int width, int height, float X0, float Y0, float Xmax, float
Ymax)
{ //инициализация графики и прорисовка осей и легенд
B = gnew Bitmap(width, height);
G = Graphics::FromImage(B);
G->SmoothingMode =
System::Drawing::Drawing2D::SmoothingMode::AntiAlias;
if (X0>Xmax)
X->Reverse=-1;
if (Y0>Ymax)
Y->Reverse=-1;
X->Beg = X0+(X->Legend_Levels*30*X->Reverse);
X->End = Xmax;
Y->Beg = Y0+(Y->Legend_Levels*30*Y->Reverse);
Y->End = Ymax;
X->Scale = System::Math::Abs(tsum/(X->End-X->Beg));
Y->Scale = System::Math::Abs(Vmax/(Y->End-Y->Beg));
Draw(0,0,tsum+10*X->Scale,0,Axes_Pen, false);
Draw(0,0,0,Vmax+10*Y->Scale,Axes_Pen, false);
Draw_String(tsum-30*X->Scale,40*Y->Scale,X-
>Symbol,Plot_Font,Plot_Brush, false);
Draw_String(5*X->Scale,Vmax,Y->Symbol,Plot_Font,Plot_Brush, false);
//Draw_String(tsum-45*X->Scale,-5*Y-
>Scale,tsum.ToString(),Plot_Font,Plot_Brush, false);
//Draw_String(5*X->Scale,Vmax-25*Y-
>Scale,Vmax.ToString(),Plot_Font,Plot_Brush, false);
Draw_String(-20*X->Scale,0,"0",Plot_Font,Plot_Brush, false);
List<Legend^>::Enumerator^ Legend_Enum = X->Legends->GetEnumerator();
while (Legend_Enum->MoveNext())
{
Legend^ lg = Legend_Enum->Current;
if (lg->Show_Level>0)
{
Draw(lg->Beg, -40*(lg->Show_Level-1)*Y->Scale-10*Y->Scale, lg->End, -
40*(lg->Show_Level-1)*Y->Scale-10*Y->Scale, Measure_Pen, false);
Draw(lg->Beg, -40*(lg->Show_Level-1)*Y->Scale-15*Y->Scale, lg->Beg,
Vmax+15*Y->Scale, Dotted_Pen, false);
Draw(lg->End, -40*(lg->Show_Level-1)*Y->Scale-15*Y->Scale, lg->End,
Vmax+15*Y->Scale, Dotted_Pen, false);
Draw_String((lg->End+lg->Beg)/2-10*X->Scale,-40*(lg->Show_Level-1)*Y-
>Scale-15*Y->Scale,lg->Symbol,Plot_Font,Plot_Brush, false);
}
}
Legend_Enum = Y->Legends->GetEnumerator();

```



```

while (Legend_Enum->MoveNext())
{
Legend^ lg = Legend_Enum->Current;
if (lg->Show_Level>0)
{
Draw(-40*(lg->Show_Level-1)*X->Scale-10*X->Scale, lg->Beg, -40*(lg->Show_Level-1)*X->Scale-10*X->Scale, lg->End, Measure_Pen, false);
Draw(-40*(lg->Show_Level-1)*X->Scale-15*X->Scale, lg->Beg, tsum+15*X->Scale,lg->Beg, Dotted_Pen, false);
Draw(-40*(lg->Show_Level-1)*X->Scale-15*X->Scale, lg->End, tsum+15*X->Scale,lg->End, Dotted_Pen, false);
Draw_String(-40*(lg->Show_Level-1)*X->Scale-40*X->Scale,(lg->End+lg->Beg)/2,lg->Symbol,Plot_Font,Plot_Brush, false);}
}
if (Draw_Object!=nullptr)
Draw_Object->Refresh();
}

```

```

public ref class Params //класс параметров (поля главного окна)
{
public:
float traz; //время разгона
float Vrab; //рабочая скорость
float ttor; //время торможения
float Vzam; //скорость замедленного хода
float tzam; //время замедленного хода
float T; //период вывода скорости на привод (соответствует параметру P
класса Plot)
Params(){ traz = 0; Vrab = 0; ttor = 0; Vzam = 0; tzam = 0; T = 0;}
};

```

```

//файл DADC_Front.h: класс формы – титульного окна программы,
выводимого при ее запуске
#pragma once

```

```

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

```

```

namespace DriveAccelDecelControl {

```

```

public ref class DADC_Front : public System::Windows::Forms::Form
{
public:
DADC_Front(void)
{
InitializeComponent();
}

protected:
~DADC_Front()
{
if (components)
{
delete components;
}
}

private: System::Windows::Forms::Button^ bHelp; //объект – кнопка
"Справка"
private: System::Windows::Forms::Button^ bRun; //объект – кнопка "Далее
>>>" запуска программы
private: System::Windows::Forms::GroupBox^ gbTitle;
private: System::Windows::Forms::Label^ lTitle;

private:
System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
/// <summary>
/// Обязательный метод для поддержки конструктора – не изменяйте
содержимое данного метода при помощи редактора кода.
/// </summary>
void InitializeComponent(void)
{
System::ComponentModel::ComponentResourceManager^ resources = (gcnew
System::ComponentModel::ComponentResourceManager(DADC_Front::typeid
));
this->bHelp = (gcnew System::Windows::Forms::Button());
this->bRun = (gcnew System::Windows::Forms::Button());
this->gbTitle = (gcnew System::Windows::Forms::GroupBox());
this->lTitle = (gcnew System::Windows::Forms::Label());
this->gbTitle->SuspendLayout();
this->SuspendLayout();

```

```

//
// bHelp
//
this->bHelp->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif",
9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->bHelp->ForeColor = System::Drawing::Color::Blue;
this->bHelp->Location = System::Drawing::Point(90, 201);
this->bHelp->Name = L"bHelp";
this->bHelp->Size = System::Drawing::Size(91, 40);
this->bHelp->TabIndex = 2;
this->bHelp->Text = L"Справка";
this->bHelp->UseVisualStyleBackColor = true;
this->bHelp->Click += gcnew System::EventHandler(this,
&DADC_Front::bHelp_Click);
//
// bRun
//
this->bRun->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif",
9.75F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->bRun->ForeColor = System::Drawing::Color::Blue;
this->bRun->Location = System::Drawing::Point(250, 201);
this->bRun->Name = L"bRun";
this->bRun->Size = System::Drawing::Size(91, 40);
this->bRun->TabIndex = 1;
this->bRun->Text = L"Далее >>>";
this->bRun->UseVisualStyleBackColor = true;
this->bRun->Click += gcnew System::EventHandler(this,
&DADC_Front::bRun_Click);
//
// gbTitle
//
this->gbTitle->Controls->Add(this->lTitle);
this->gbTitle->Location = System::Drawing::Point(9, 7);
this->gbTitle->Name = L"gbTitle";
this->gbTitle->Size = System::Drawing::Size(404, 180);
this->gbTitle->TabIndex = 3;
this->gbTitle->TabStop = false;
//
// lTitle
//

```

```

this->ITitle->AutoSize = true;
this->ITitle->Dock = System::Windows::Forms::DockStyle::Fill;
this->ITitle->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif",
17.25F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->ITitle->ForeColor = System::Drawing::Color::Red;
this->ITitle->Image = (cli::safe_cast<System::Drawing::Image^ >(resources-
>GetObject(L"ITitle.Image")));
this->ITitle->ImageAlign = System::Drawing::ContentAlignment::BottomRight;
this->ITitle->Location = System::Drawing::Point(3, 16);
this->ITitle->Name = L"ITitle";
this->ITitle->Size = System::Drawing::Size(400, 145);
this->ITitle->TabIndex = 0;
this->ITitle->Text = L"Система разгона-торможения \нуправляемого
привода \н\nАвторы: Я.Л. Либерман \н      "
L"      К.Ю. Летнев";
//
// DADC_Front
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(423, 257);
this->ControlBox = false;
this->Controls->Add(this->gbTitle);
this->Controls->Add(this->bRun);
this->Controls->Add(this->bHelp);
this->MaximizeBox = false;
this->MinimizeBox = false;
this->Name = L"DADC_Front";
this->ShowInTaskbar = false;
this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
this->gbTitle->ResumeLayout(false);
this->gbTitle->PerformLayout();
this->ResumeLayout(false);

}
#pragma endregion

private: System::Void bHelp_Click(System::Object^ sender,
System::EventArgs^ e) {
    Form^ dadc_help = gcnew DADC_Help(); //при нажатии на кнопку
    "Справка" создаем окно справки

```

```

dadc_help->Show();
}
private: System::Void bRun_Click(System::Object^ sender,
System::EventArgs^ e) {
    this->Hide(); //при нажатии на кнопку "Далее >>>" скрываем это окно
    Form^ dadc = gcnew DADC(); //и создаем и показываем главное окно
    dadc->Show();
}
};
}

//файл DADC_Help.h: класс формы – окна справки
#pragma once

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

namespace DriveAccelDecelControl {

public ref class DADC_Help : public System::Windows::Forms::Form
{
public:
    DADC_Help(void)
    {
        InitializeComponent();
        DADC_Help_File = "Инструкция.rtf"; //файл со справкой
        rtbHelp->LoadFile(DADC_Help_File); //и его загрузка при создании окна
    }

protected:
    ~DADC_Help()
    {
        if (components)
        {
            delete components;
        }
    }
private: System::Windows::Forms::RichTextBox^ rtbHelp;

private: System::String^ DADC_Help_File;

```

protected:

private:

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Обязательный метод для поддержки конструктора – не изменяйте
содержимое данного метода при помощи редактора кода

/// </summary>

void InitializeComponent(void)

{

this->rtbHelp = (gcnew System::Windows::Forms::RichTextBox());

this->SuspendLayout();

//

// rtbHelp

//

this->rtbHelp->Dock = System::Windows::Forms::DockStyle::Fill;

this->rtbHelp->Location = System::Drawing::Point(0, 0);

this->rtbHelp->Name = L"rtbHelp";

this->rtbHelp->Size = System::Drawing::Size(502, 383);

this->rtbHelp->TabIndex = 0;

this->rtbHelp->Text = L"";

//

// DADC_Help

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(502, 383);

this->Controls->Add(this->rtbHelp);

this->Name = L"DADC_Help";

this->StartPosition =

System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Справка системы разгона–торможения управляемого
привода";

this->ResumeLayout(false);

}

#pragma endregion

};

}

Научное издание

Либерман Яков Львович

Летнев Константин Юрьевич

**Автоматическое устранение колебаний груза
на гибкой подвеске при подъеме и опускании**

Редактор *О. В. Байгулова*

Компьютерный набор *К. Ю. Летнева*

Подписано в печать 03.07.2012.
Формат 60x84 1/16. Бумага писчая.
Плоская печать. Усл. печ. л. 12,47.
Уч.-изд. л. 10,4. Тираж 150 экз. Заказ

Редакционно-издательский отдел УрФУ
620002, Екатеринбург, ул. Мира, 19